

DEVICE AND METHOD FOR IMPLEMENTING INTERFACE AT HIGH DATA TRANSFER SPEED

Publication number: RU2337497 (C2)

Publication date: 2008-10-27

Inventor(s): ANDERSON DZHON DZHEJMS [US]; STIL BRAJAN [US]; UAJLI DZHORDZH A [US]; SHEKKHAR SHASHANK [US] + (ANDERSON DZHON DZHEJMS, ; STIL BRAJAN, ; UAJLI DZHORDZH A, ; SHEKKHAR SHASHANK)

Applicant(s): QUALCOMM INC [US] + (KVEHLKOMM INKORPOREJTED)

Classification:

- international: H04L12/28; H04L12/56; H04L29/06; H04L12/28; H04L12/56; H04L29/06

- European: H04L29/06C2; H04L29/06K; H04L29/06N

Application number: RU20060135633 20050310

Priority number(s): US20040552176P 20040310; US20040554309P 20040317

Abstract of RU 2337497 (C2)

FIELD: physics, communications. ^ SUBSTANCE: invention concerns digital signal transfer and claims data transfer interface for digital data transfer between host and user over data transfer path using packet structures linked together to form data transfer protocol, for exchange of pre-selected digital control and display data set. Signal transfer protocol is used by data transfer line controllers configured to generate, receive and send packets forming data transfer protocol and to form digital data packets of one or several types, so that at least one is located at host device and is connected to user by data transfer path.; Interface enables economy, low-energy, two-way high-speed data transfer mechanism over data transfer line of sequential type with narrow operation range, used with mini couplers and thin flexible cables especially efficient for connection of display elements, such as portable microdisplays, to portable computers and wireless data exchange devices. ^ EFFECT: higher data transfer speed. ^ 45 cl, 117 dwg, 20 tbl



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: 2006135633/09, 10.03.2005

(24) Дата начала отсчета срока действия патента:
10.03.2005

(30) Конвенционный приоритет:
10.03.2004 US 60/552,176

(43) Дата публикации заявки: 20.04.2008

(45) Опубликовано: 27.10.2008 Бюл. № 30

(56) Список документов, цитированных в отчете о
поиске: WO 2003/023587, A2, 20.03.2003. RU
2134447, C2, 10.08.1999. US 5751951, A,
12.05.1998.

(85) Дата перевода заявки РСТ на национальную фазу:
10.10.2006

(86) Заявка РСТ:
US 2005/008073 (10.03.2005)

(87) Публикация РСТ:
WO 2005/088939 (22.09.2005)

Адрес для переписки:
129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(72) Автор(ы):

АНДЕРСОН Джон Джеймс (US),
СТИЛ Брайан (US),
УАЙЛИ Джордж А. (US),
ШЕКХАР Шашанк (US)

(73) Патентообладатель(и):

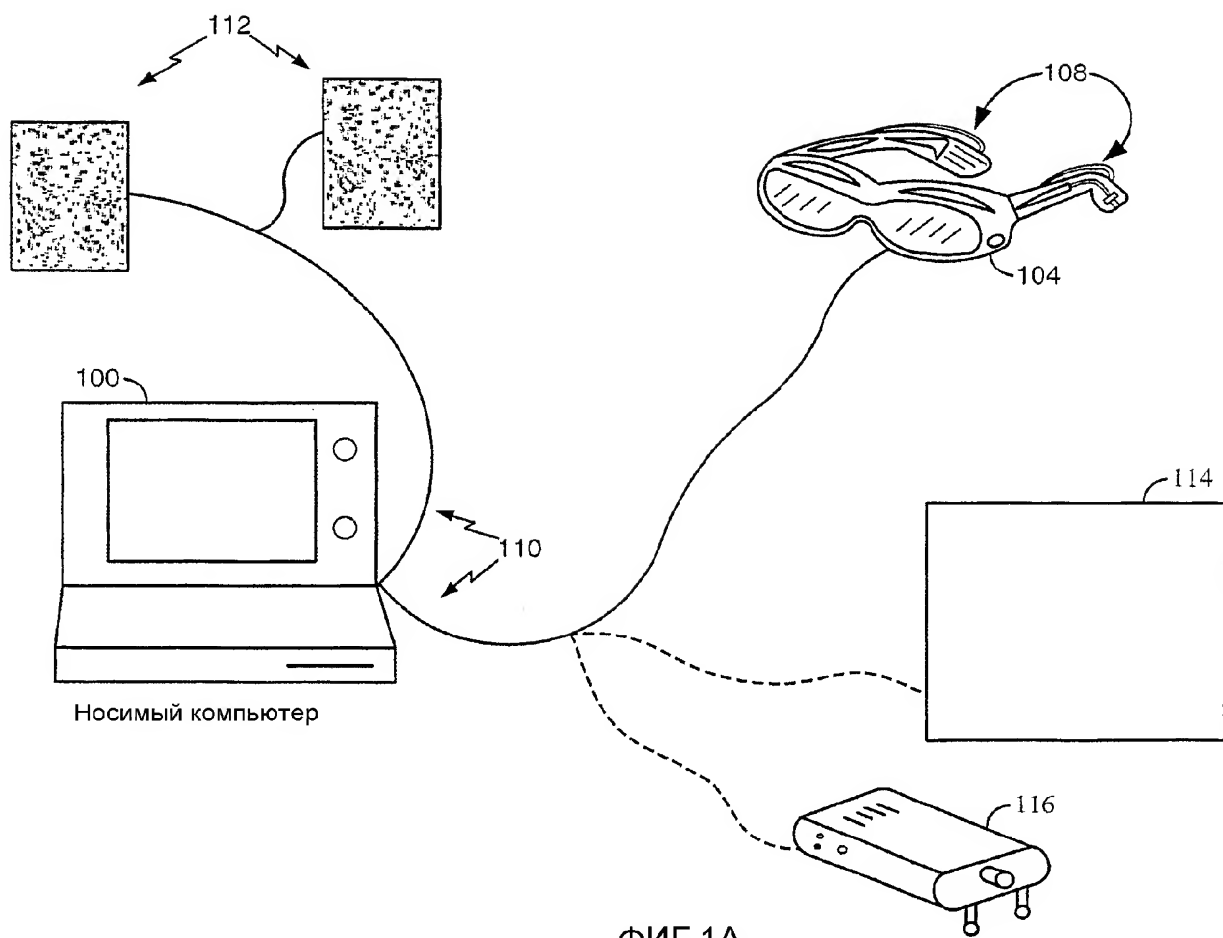
КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(54) УСТРОЙСТВО И СПОСОБ ДЛЯ РЕАЛИЗАЦИИ ИНТЕРФЕЙСА С ВЫСОКОЙ СКОРОСТЬЮ ПЕРЕДАЧИ ДАННЫХ

(57) Реферат:

Изобретение относится к передаче цифровых сигналов. Технический результат - повышение скорости передачи данных. Интерфейс передачи данных для пересылки цифровых данных между хостом и клиентом по тракту обмена данными, используя структуры пакетов, связанные вместе для формирования протокола обмена данными, для обмена предварительно выбранным набором цифровых данных управления и представления. Протокол передачи сигналов используется контроллерами линии передачи данных, сконфигурированными для генерирования, передачи и приема пакетов, формирующих протокол обмена данными, и для формирования

цифровых данных в пакеты данных одного или нескольких типов, причем, по меньшей мере, один находится в устройстве хоста и связан с клиентом при помощи тракта обмена данными. Интерфейс обеспечивает экономичный маломощный двунаправленный механизм пересылки данных с высокой скоростью по линии передачи данных «последовательного» типа с малым радиусом действия, который применяется для реализации с миниатюрными соединителями и тонкими гибкими кабелями, которые особенно полезны при подключении дисплейных элементов, таких как носимые микродисплеи, к портативным компьютерам и беспроводным устройствам обмена данными. 12 н. и 33 з.п. ф-лы, 97 ил., 20 табл.



ФИГ.1А



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION(21), (22) Application: **2006135633/09, 10.03.2005**(24) Effective date for property rights: **10.03.2005**(30) Priority:
10.03.2004 US 60/552,176(43) Application published: **20.04.2008**(45) Date of publication: **27.10.2008 Bull. 30**(85) Commencement of national phase: **10.10.2006**(86) PCT application:
US 2005/008073 (10.03.2005)(87) PCT publication:
WO 2005/088939 (22.09.2005)

Mail address:
**129090, Moskva, ul. B.Spasskaja, 25, str.3,
OOO "Juridicheskaja firma Gorodisskij i
Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595**

(72) Inventor(s):

**ANDERSON Dzhon Dzhejms (US),
STIL Brajan (US),
UAJLI Dzhordzh A. (US),
ShEKKhAR Shashank (US)**

(73) Proprietor(s):

KVEhKOMM INKORPOREJTED (US)

(54) DEVICE AND METHOD FOR IMPLEMENTING INTERFACE AT HIGH DATA TRANSFER SPEED

(57) Abstract:

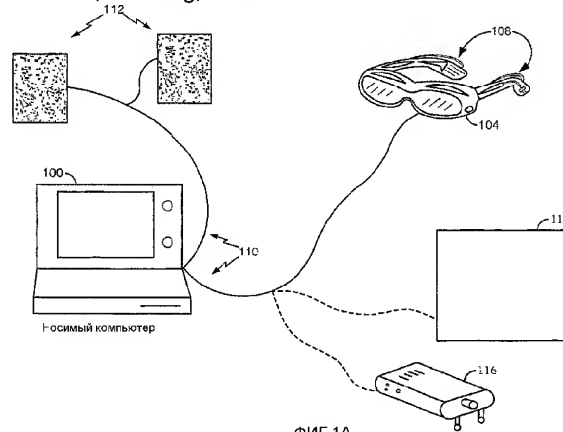
FIELD: physics, communications.

SUBSTANCE: invention concerns digital signal transfer and claims data transfer interface for digital data transfer between host and user over data transfer path using packet structures linked together to form data transfer protocol, for exchange of pre-selected digital control and display data set. Signal transfer protocol is used by data transfer line controllers configured to generate, receive and send packets forming data transfer protocol and to form digital data packets of one or several types, so that at least one is located at host device and is connected to user by data transfer path. Interface enables economy, low-energy, two-way high-speed data transfer mechanism over data transfer line of sequential type with narrow operation range, used with mini couplers and thin flexible cables especially efficient for connection of display

elements, such as portable microdisplays, to portable computers and wireless data exchange devices.

EFFECT: higher data transfer speed.

45 cl, 117 dwg, 20 tbl



Притязание на приоритет по §119 раздела 35Кодекса законов США

По настоящей заявке на патент испрашивается приоритет предварительной заявки № 60/552176, озаглавленной «Switchable Threshold Differential Interface»

- 5 (Дифференциальный интерфейс с переключаемым порогом), поданной 10 марта 2004 г., и предварительной заявки № 60/554309, озаглавленной «Switchable Threshold Differential Interface» (Дифференциальный интерфейс с переключаемым порогом), поданной 17 марта 2004 г., причем правопреемником обеих заявок является правопреемник данной заявки, и они настоящим документом специально включены в данный документ посредством ссылки.

10 **Область техники, к которой относится изобретение**

- Варианты осуществления изобретения в данном раскрытии относятся к протоколу, процессу и устройству передачи цифровых сигналов, включая интегральные схемы и компоненты, для передачи или пересылки сигналов между устройством хоста и устройством клиента с высокими скоростями передачи данных. Более конкретно раскрытие
- 15 относится к методу передачи цифровых сигналов мультимедиа и других типов с устройства хоста или контроллера на устройство клиента для представления или отображения конечному пользователю, используя механизм передачи с малой потребляемой мощностью и высокой скоростью передачи данных, имеющий внутренние и внешние применения устройств.

20 **Предшествующий уровень техники**

- Последние несколько лет наблюдается значительный прогресс в компьютерах, относящихся к электронным играм продуктах и различных видеотехнологиях (например, цифровые многофункциональные диски (ЦМД) и кассетные видеомagnetофоны высокой четкости) для выполнения представления фотоснимков, видео, видео по требованию и
- 25 графических изображений со все более высоким разрешением, даже включая некоторые типы текстов, конечным пользователям такого оборудования. Такой прогресс, в свою очередь, давал право использовать электронные устройства для просмотра с более высоким разрешением, такие как видеомониторы высокой четкости, мониторы телевидения высокой четкости (ТВЧ) или специализированные элементы проекции изображения.
- 30 Объединение таких визуальных изображений с аудиоданными с высоким разрешением или высококачественными аудиоданными, такими как при использовании воспроизведение звука с компакт-дисков, ЦМД, с объемным звучанием, и другими устройствами, также имеющими связанные с ними выходы аудиосигналов, используется для создания более реалистичного, насыщенного содержимым или истинным мультимедийным впечатлением
- 35 для конечного пользователя. Кроме того, звуковые системы и механизмы транспортировки музыки с высокой степенью мобильности и высоким качеством, такие как проигрыватели формата МР3 (формат сжатия звука уровня 3 Экспертной группы по движущимся изображениям (ЭГДИ)), были разработаны только для представления звука конечным пользователям. Это привело к растущим ожиданиям для обычных пользователей
- 40 коммерческих электронных устройств, от компьютеров до телевидения и даже телефонов, привыкшим в настоящее время и ожидающим результат с высоким или отличным качеством.

- В обычном сценарии представления видео, включающим в себя электронный продукт, видеоданные обычно пересылаются с использованием настоящих технологий со скоростью
- 45 передачи, которая наилучшим образом может быть охарактеризована как медленная или средняя будучи порядка от одного до десятка килобит в секунду. Эти данные затем или буферизуются, или запоминаются во временных или долговременных запоминающих устройствах для задержанного (более позднего) воспроизведения на требуемом устройстве просмотра. Например, изображения могут пересылаться «по» или используя интернет,
- 50 используя программу, находящуюся резидентно на компьютере, имеющем модем или устройство подключения к интернету другого типа, для приема или передачи данных, используемых при цифровом представлении изображения. Подобная передача может иметь место с использованием беспроводных устройств, таких как портативные

компьютеры, оснащенные беспроводными модемами, или беспроводных персональных цифровых помощников (ПЦП) или беспроводных телефонов.

Если данные приняты, они хранятся локально в элементах, схемах или устройствах памяти, таких как оперативное запоминающее устройство (ОЗУ) или флэш-память, включая
 5 внутренние или внешние запоминающие устройства, такие как накопители на жестких дисках малого размера, для воспроизведения. В зависимости от количества данных и разрешения изображения воспроизведение может начинаться относительно быстро или может представляться с длительной задержкой. Т.е. в некоторых случаях представление изображения предусматривает некоторую степень воспроизведения в реальном времени
 10 для изображений с очень малым или низким разрешением, не требующим большого количества данных или использующим буферизацию некоторого типа, так что после небольшой задержки некоторые данные представляются, в то время как дополнительные данные пересылаются. При условии, что нет перерывов на линии передачи данных или помех от других систем или пользователей в отношении используемой линии передачи,
 15 если представление началось, пересылка является достаточно прозрачной для конечного пользователя устройства просмотра. Конечно, если многочисленные пользователи совместно используют один тракт обмена данными, такой как проводное подключение к интернету, пересылки могут прерываться или замедляться по сравнению с желаемым.

Данные, используемые для создания или неподвижных изображений, или движущегося
 20 видео, часто сжимаются с использованием одного из нескольких общеизвестных методов, таких как те, которые определены Объединенной группой экспертов по фотографическим изображениям (ОГЭФИ), ЭГДИ и другими общеизвестными организациями или компаниями по стандартизации в отраслях систем связи, компьютерной техники и передачи данных, для ускорения передачи данных по линии передачи данных. Это позволяет передавать
 25 изображения или данные быстрее посредством использования меньшего количества битов для передачи данного количества информации.

Если данные пересылаются на «локальное» устройство, такое как компьютер, имеющий механизм хранения, такой как память, или магнитные или оптические запоминающие элементы, или на другие приемные устройства, устраняется сжатие результирующей
 30 информации (или результирующая информация воспроизводится с использованием специальных декодирующих проигрывателей), и она декодируется, если необходимо, и готовится для соответствующего представления, основываясь на соответствующем доступном разрешении представления и элементах управления. Например, типовое разрешение видеосистемы компьютера в единицах разрешения экрана X на Y пикселей
 35 обычно находится в диапазоне от минимального 480×640 пикселей, с промежуточным 600×800 и до 1024×1024, хотя, по существу, возможны многочисленные другие разрешения, или требуемые, или необходимые.

На представление изображения также оказывает влияние содержимое изображения и возможность данных видеоконтроллеров манипулировать изображением в понятиях
 40 некоторых предварительно определенных уровней цвета или глубины цвета (битов на пиксель, используемых для генерирования цвета) и яркости и любых дополнительных используемых служебных битах. Например, типовое представление на компьютере предусматривает что-то от примерно 8 до 32 или более битов на пиксель для представления разнообразных цветов (тонов и оттенков), хотя встречаются другие
 45 значения.

Из вышеупомянутых значений можно видеть, что данное изображение экрана потребует передачу что-то от 2,45 мегабитов (Мбит) до примерно 33,55 Мбит данных в интервале от наименьшего до наибольшего типового разрешения и глубины соответственно. При просмотре видео или изображений движущегося типа с частотой 30 кадров в секунду
 50 количество необходимых данных составляет примерно от 73,7 до 1006 мегабитов данных в секунду (Мбит/с) или примерно от 9,21 до 125,75 мегабайтов в секунду (Мбайт/с). Кроме того, кому-то может потребоваться представление аудиоданных совместно с изображениями, например, для представления средствами мультимедиа, или отдельного

представления аудио с высоким разрешением, такого как музыка качества компакт-диска. Также могут использоваться дополнительные сигналы, имеющие дело с интерактивными командами, элементами управления или сигналами. Каждый из этих вариантов добавляет еще больше данных, подлежащих пересылке. Кроме того, более современные методы

5 передачи, включающие в себя телевидение высокой четкости (ВЧ) и записи фильмов, могут добавлять еще больше данных и информации управления. В любом случае, когда требуется пересылать данные изображения с высоким качеством или с высоким разрешением и высококачественную аудиоинформацию или сигналы данных конечному

10 пользователю для создания богатого содержимым впечатления, требуется линия передачи данных с высокой скоростью передачи данных между элементами представления и источником или устройством хоста, который конфигурируется для предоставления данных таких типов.

Скорости передачи данных примерно 115 килобайтов (кбайт/с) или 920 килобитов в секунду (кбит/с) могут в обычном порядке обрабатываться некоторыми современными

15 последовательными интерфейсами. Другие интерфейсы, такие как последовательные интерфейсы универсальной последовательной шины (УПШ), могут обеспечивать пересылку данных с максимальной скоростью 12 Мбайт/с, и специализированные высокоскоростные пересылки, такие как те, которые конфигурируются с использованием стандарта 1394 Института инженеров по электротехнике и радиоэлектронике (ИИЭР), могут

20 происходить со скоростями передачи порядка 100-400 Мбайт/с. К сожалению, эти скорости передачи оказываются недостаточными для требуемых высоких скоростей передачи данных, описанных выше, которые рассматриваются для использования с будущими беспроводными устройствами передачи данных или другими службами для предоставления выходных сигналов с богатым содержимым и с высоким разрешением для

25 работы портативных видеодисплеев или звуковых устройств. Они включают в себя компьютеры для бизнеса и других представлений, игровые устройства и т.п. Кроме того, эти интерфейсы требуют использования для работы значительного количества программного обеспечения для хоста или системы и клиента. Их стеки протоколов программного обеспечения также создают нежелательное большое количество служебной

30 информации, особенно там, где рассматриваются применения мобильных беспроводных устройств или телефонов. Такие устройства имеют жесткие ограничения на память и потребление мощности, а также уже облагаемые налогом вычислительные возможности. Кроме того, некоторые из этих интерфейсов используют громоздкие кабели, которые являются слишком тяжелыми и неудовлетворительными для мобильных применений,

35 ориентированных на высокую эстетику, сложные соединители, которые добавляют стоимость, или просто потребляют слишком большую мощность.

Существуют другие известные интерфейсы, такие как интерфейс аналогового видеографического адаптера (АВГА), интерактивный цифровой видео (ИЦВ) или интерфейс к видеоканалу Гигабит (ИВГ). Первые два из них являются интерфейсами параллельного

40 типа, которые обрабатывают данные при более высоких скоростях передачи, но также используют тяжелые кабели и потребляют большую мощность, порядка нескольких ватт. Ни одна из этих характеристик не подлежит применению с портативными бытовыми электронными устройствами. Даже третий интерфейс потребляет слишком большую мощность и использует дорогие или громоздкие соединители.

45 Для некоторых из вышеописанных интерфейсов и других систем/протоколов передачи с очень высокой скоростью передачи данных или механизмов передачи, связанных с передачами данных для стационарного компьютерного оборудования, существует другой существенный недостаток. Чтобы обеспечить требуемые скорости передачи данных, также требуется существенное количество мощности и/или работа при высоких уровнях тока. Это

50 значительно снижает полезность таких методов для ориентированных на потребителя продуктов с очень высокой мобильностью.

В основном, чтобы обеспечить такие скорости пересылки данных, используя альтернативные решения, такие как, например, подключения и элементы передачи опτικο-

волоконного типа, требуется также ряд дополнительных преобразователей и элементов, которые вносят значительно больше сложности и затрат, чем требуется для действительно коммерческого, ориентированного на потребителя продукта. Помимо в основном дорогой сущности оптических систем на данный момент их потребляемая мощность и сложность

5 предотвращают обычное использование для легковесных, маломощных и портативных применений.

В промышленности для портативных, беспроводных или мобильных применений отсутствовала методика обеспечения опыта высококачественного представления, основывается ли оно на аудио, видео или мультимедиа, для в высокой степени мобильных

10 конечных пользователей. Т.е. при использовании портативных компьютеров, беспроводных телефонов, ПЦП или других в высокой степени мобильных устройств или оборудования обмена данными настоящие просто используемые системы или устройства представления видео и аудио не могут предоставить выходной результат на требуемом

15 высококачественном уровне. Часто воспринимаемое качество, которого не хватает, является результатом недоступных высоких скоростей передачи данных, требуемых для пересылки высококачественных данных представления. Это может включать в себя как пересылку на более эффективные, усовершенствованные или нагруженные

20 возможностями внешние устройства для представления конечному пользователю, так и пересылку между хостами и клиентами, внутренними для портативных устройств, таких как компьютеры, игровые автоматы, и беспроводными устройствами, такими как мобильные телефоны.

В этом последнем случае были достигнуты большие успехи в добавлении внутренних видеоз экранов со все более и более высоким разрешением и других специальных устройств ввода и/или вывода и подключений к беспроводным устройствам подобно так называемым

25 телефонам третьего поколения и так называемым носимым компьютерам. Однако внутренние шины данных и подключения, которые могут включать в себя замыкание контактов через вращающиеся или скользящие шарниры или подобные шарнирам конструкции, которые устанавливают или подсоединяют видеоз экраны или другие элементы к основному корпусу, где постоянно находятся хост и/или другие различные элементы

30 управления и выходные компоненты, представляют собой в основном интерфейсы с большой полосой пропускания или высокой пропускной способностью. Очень трудно создать интерфейсы передачи данных с высокой пропускной способностью, используя известные методики, которые могут потребовать до 90 проводников или более для достижения требуемой пропускной способности, например, в качестве одного примера, на

35 беспроводном телефоне. Настоящие решения обычно включают в себя использование интерфейсов параллельного типа с относительно высокими уровнями сигналов, которые могут вызывать то, что межсоединения являются более дорогими, менее надежными и потенциально могут генерировать излучения, которые могут создавать помехи функционированию устройства. Это представляет многие требующие решения проблемы,

40 подвергающие сомнению производство, стоимость и надежность.

Такие проблемы и требования также наблюдаются на стационарных установках, где устройства обмена данными или вычислительного типа в качестве одного примера добавляются к приборам и другим бытовым устройствам для предоставления

45 усовершенствованных возможностей передачи данных, подключения к интернету и подключения для передачи данных или встроенных развлекательных возможностей. Другим примером являются самолеты и автобусы, где индивидуальные экраны представления видео и аудио вмонтированы в спинки сидений. Однако в этих ситуациях часто является более удобным, эффективным и легко обслуживаемым иметь основные элементы управления хранением, обработкой или обмена данными, расположенные на

50 расстоянии от видимых экранов или выходов аудио с соединительной линией передачи данных или каналом для представления информации. Эта линия передачи данных потребует манипулирование значительным количеством данных для достижения требуемой пропускной способности, как описано выше.

Поэтому требуется новый механизм пересылки для повышения пропускной способности передачи данных между устройствами хоста, предоставляющими данные, и устройствами или элементами дисплея клиента, представляющими выходной результат конечным пользователям.

- 5 Заявители предложили такие новые механизмы пересылки в заявке на патент США № 10/020520, поданной 14 декабря 2001 г., теперь патент США № 6760772, выданный 6 июля 2004 г. Zou et al., и в заявке на патент США № 10/236657, поданной 6 сентября 2002 г., при этом обе озаглавлены «Generating And Implementing A Communication Protocol And Interface For High Data Rate Signal Transfer» (Генерирование и реализация
- 10 протокола и интерфейса обмена данными для передачи сигналов с высокой скоростью передачи данных), теперь признанные патентоспособными, правопреемником которых является правопреемник настоящего изобретения и которые включены в данный документ по ссылке. Также заявка США № 10/860116, поданная 2 июня 2004 г., озаглавленная «Generating and Implementing a Signal Protocol and Interface for Higher Data Rates»
- 15 (Генерирование и реализация протокола и интерфейса передачи сигналов для более высоких скоростей передачи данных). Методики, описанные в этих заявках, могут существенно повысить скорость передачи для больших количеств данных в высокоскоростных сигналах данных. Однако продолжает расти спрос на все возрастающие скорости передачи данных, особенно относящиеся к представлениям видео. Даже с
- 20 другими ведущимися разработками в технологии передачи сигналов данных все еще существует потребность достижения еще более высоких скоростей передачи, повышенных эффективностей линии обмена данными и более мощных линий обмена данными. Поэтому существует продолжающаяся потребность в разработке нового или улучшенного механизма пересылки, который требуется для повышения пропускной способности данных,
- 25 между устройствами хоста и клиента.

Сущность изобретения

- Варианты осуществления изобретения касаются вышеупомянутого недостатка и других существующих в известном уровне техники, в которых был разработан новый протокол и средство, способ и механизм пересылки данных для пересылки данных между устройством
- 30 хоста и приемным устройством клиента с высокими скоростями передачи данных.

- Варианты осуществления для изобретения относятся к мобильному интерфейсу цифровых данных (МИЦД) для пересылки цифровых данных с высокой скоростью передачи между устройством хоста и устройством клиента по тракту обмена данными, который
- 35 использует множество или группу структур пакетов для формирования протокола обмена данными для обмена предварительно выбранным набором цифровых данных управления и представления между устройствами хоста и клиента. Протокол передачи сигналов или канальный уровень используется физическим уровнем контроллеров линий передачи данных, приемников или драйверов хоста или клиента. По меньшей мере один контроллер или драйвер линии передачи данных, находящийся в устройстве хоста, связывается с
- 40 устройством клиента по тракту обмена данными или линии передачи данных и конфигурируется для генерирования, передачи и приема пакетов, формирующих протокол обмена данными, и для формирования цифровых данных представления в пакеты данных одного или несколько типов. Интерфейс обеспечивает двунаправленную пересылку информации между хостом и клиентом, который может постоянно находиться в общем
- 45 корпусе или опорной конструкции.

- Реализация в основном вся цифровая по своей сущности за исключением дифференциальных драйверов и приемников, которые легко могут быть реализованы на цифровом кристалле комплементарной структуры «металл-оксид-полупроводник» (КМОП),
- 50 требует всего 6 сигналов и работает почти на любой скорости передачи данных, которая удобна для разработчика системы. Простой протокол физического и канального уровней делает легкой ее интеграцию, и эта простота плюс состояние «спячки» делают возможным для портативных систем иметь очень низкое потребление мощности системой.

Чтобы способствовать использованию и признанию, интерфейс добавляет очень мало к

стоимости устройства, предусматривает потребление очень малой мощности, в то же время может обеспечивать питание дисплея через интерфейс, используя стандартные напряжения батарей, и может обеспечивать устройства, имеющие карманный форм-фактор. Интерфейс является масштабируемым для поддержки разрешений свыше ТВЧ, поддерживает одновременное стерео видео и аудио по схеме 7.1 для устройства дисплея, выполняет условное обновление любой области экрана и поддерживает многочисленные типы данных в обоих направлениях.

В других аспектах вариантов осуществления изобретения по меньшей мере один контроллер линии передачи данных, приемник, устройство или драйвер клиента расположен в устройстве клиента и связывается с устройством хоста по тракту обмена данными или линии передачи данных. Контроллер линии передачи данных клиента также конфигурируется для генерирования, передачи и приема пакетов, формирующих протокол обмена данными, и для формирования цифровых данных представления в пакеты данных одного или нескольких типов. Как правило, контроллер хоста или линии передачи данных использует конечный автомат для обработки пакетов данных, используемых в командах или некоторых типах подготовки сигналов, и обработки запросов, но может использовать более медленный процессор общего назначения для манипулирования данными и некоторыми менее сложными пакетами, используемыми в протоколе обмена данными. Контроллер хоста содержит один или несколько дифференциальных линейных драйверов; тогда как приемник клиента содержит один или несколько дифференциальных линейных приемников, связанных с трактом обмена данными.

Пакеты группируются вместе в кадры полезной информации (медиакадры), обмен которыми осуществляется между устройствами хоста и клиента, имеющие предварительно определенную фиксированную длину с предварительно определенным количеством пакетов, имеющих различные переменные длины. Каждый пакет содержит поле длины пакета, одно или несколько полей данных пакета и поле циклического избыточного кода. Пакет заголовка подкадра пересылается или располагается в начале пересылки других пакетов от контроллера линии передачи данных хоста. Один или несколько пакетов типа Видеопотока и пакетов типа Аудиопотока используются протоколом обмена данными для пересылки данных типа видео и данных типа аудио соответственно с хоста на клиент по прямой линии передачи данных для представления пользователю устройства клиента. Один или несколько пакетов типа Инкапсуляции обратной линии передачи данных используются протоколом обмена данными для пересылки данных с устройства клиента на контроллер линии передачи данных хоста. Эта пересылка в некоторых вариантах осуществления включает в себя пересылку данных с внутренних контроллеров, имеющих по меньшей мере одно устройство МИЦД, на внутренние видеоз экраны. Другие варианты осуществления включают в себя пересылку на внутренние звуковые системы и пересылку с различных устройств ввода, включающих в себя джойстики и комплексные клавиатуры, на внутренние устройства хоста.

Пакеты типа заполнителя генерируются контроллером линии передачи данных хоста и занимают периоды передачи прямой линии передачи данных, которые не имеют данных. Множество других пакетов используется протоколом обмена данными для пересылки видеоинформации. Такие пакеты включают в себя пакеты типа Карта цветов, Пересылка битовых блоков, Заливка растровой области, Заливка растровым узором и Разрешение прозрачного цвета. Пакеты типа Определяемого пользователем потока используются протоколом обмена данными для пересылки данных определенного пользователем интерфейса. Пакеты типа Данные клавиатуры и Данные указательного устройства используются протоколом обмена данными для пересылки данных на и с устройств пользовательского ввода, связанных с упомянутым устройством клиента. Пакет типа Закрытие линии передачи данных используется протоколом обмена данными для завершения пересылки данных в любом направлении по упомянутому тракту обмена данными.

Пакеты Состояния питания дисплея генерируются для обеспечения структуры, средства

или способа для перевода заданных аппаратных средств контроллера дисплея в состояние малого потребления мощности, когда клиент, такой как дисплей, не используется, или в текущем активном использовании, чтобы минимизировать потребление мощности системой или истощение системных ресурсов. В одном варианте осуществления клиент указывает

5 возможность ответа на Пакеты состояния питания дисплея, используя бит 9 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента.

Формат одного варианта осуществления для Пакета состояния питания дисплея, этот тип пакета структурируется так, что имеет поля Длины пакета, Типа пакета, Идентификатора (ИД) hClient, Состояния питания и Циклического избыточного кода (ЦИК).

10 Этот тип пакета обычно идентифицируется как пакет Типа 75 в 2-байтовом поле типа. 2-байтовое поле ИД hClient содержит информацию или значения, которые резервируются для ИД клиента. Поле состояния питания определяет информацию, используемую для перевода конкретного дисплея в заданное состояние питания в соответствии со значением некоторых предварительно выбранных битов. 2-байтовое поле ЦИК определяет или

15 содержит ЦИК всех байтов в пакете, включая Длину пакета.

Тракт обмена данными обычно содержит или использует кабель, имеющий группу из четырех или более проводников и экран. Кроме того, могут использоваться печатные провода или проводники, когда требуется, при этом некоторые находятся на гибких подложках.

20 Контроллер линии передачи данных хоста запрашивает информацию о возможностях дисплея от устройства клиента, чтобы определить, какой тип данных и какие скорости передачи данных упомянутый клиент может обеспечивать при помощи упомянутого интерфейса. Контроллер линии передачи данных клиента передает возможности дисплея или представления на контроллер линии передачи данных хоста, используя по меньшей

25 мере один пакет типа Возможностей клиента. Многочисленные режимы пересылки используются протоколом обмена данными, причем каждый позволяет выполнять пересылку различного максимального количества битов данных в параллельной форме в течение данного периода времени, каждый режим может выбираться по согласованию между контроллерами линии передачи данных хоста и клиента. Эти режимы пересылки

30 являются динамически регулируемые во время пересылки данных, и на обратной линии передачи данных необязательно должен использоваться режим, одинаковый с режимом, который используется на прямой линии передачи данных.

В других аспектах некоторых вариантов осуществления изобретения устройство хоста содержит беспроводное устройство обмена данными, такое как беспроводный телефон,

35 беспроводный ПЦП или портативный компьютер, имеющий расположенный в нем беспроводный модем. Обычное устройство клиента содержит портативный видеодисплей, такой как устройство микродисплея, и/или портативную систему представления аудио. Кроме того, хост может использовать запоминающее средство или элементы для хранения данных представления или мультимедиа, подлежащих пересылке для представления

40 пользователю устройства клиента.

В еще других аспектах некоторых вариантов осуществления устройство хоста содержит контроллер или устройство управления линией обмена данными с драйверами, как описано ниже, находящееся в портативном электронном устройстве, таком как беспроводное устройство обмена данными, такое как беспроводный телефон, беспроводный ПЦП или

45 портативный компьютер. Обычное устройство клиента в данной конфигурации содержит схему или интегральную схему клиента или модуль, связанный с хостом и находящийся в этом же устройстве и с внутренним видеодисплеем, таким как экран высокого разрешения для мобильного телефона, и/или портативную систему представления аудио, или альтернативно некоторый тип системы или устройства ввода.

50 Перечень фигур чертежей

Дополнительные признаки и преимущества, а также конструкция и принцип действия различных вариантов осуществления изобретения описываются подробно ниже с ссылкой на прилагаемые чертежи. На чертежах подобные позиции в основном обозначают

идентичные, функционально подобные и/или конструктивно подобные элементы или этапы обработки.

Фиг.1А иллюстрирует базовую среду, в которой могут работать варианты осуществления изобретения, включая использование устройства микродисплея, или проектор, совместно с портативным компьютером или другим устройством обработки данных.

Фиг.1В иллюстрирует базовую среду, в которой могут работать варианты осуществления изобретения, включая использование устройства микродисплея или проектора и элементов представления аудио, используемых совместно с беспроводным приемопередатчиком.

Фиг.2А иллюстрирует базовую среду, в которой могут работать варианты осуществления изобретения, включая использование внутреннего дисплея или устройств представления аудио, используемых в портативном компьютере.

Фиг.2В иллюстрирует базовую среду, в которой могут работать варианты осуществления изобретения, включая использование внутреннего дисплея или элементов представления аудио, используемых в беспроводном приемопередатчике.

Фиг.3 иллюстрирует общий принцип мобильного интерфейса цифровых данных со взаимным подключением хоста и клиента.

Фиг.4 иллюстрирует структуру пакета, используемую для реализации пересылки данных с устройства клиента на устройство хоста.

Фиг.5 иллюстрирует использование контроллера линии передачи данных МИЦД и типы сигналов, проходящих между хостом и клиентом по физическим проводникам линии передачи данных для интерфейса Типа 1.

Фиг.6 иллюстрирует использование контроллера линии передачи данных МИЦД и типы сигналов, проходящих между хостом и клиентом по физическим проводникам линии передачи данных для интерфейсов Типа 2, 3 и 4.

Фиг.7 иллюстрирует структуру кадров и подкадров, используемых для реализации протокола интерфейса.

Фиг.8 иллюстрирует общую структуру пакетов, используемых для реализации протокола интерфейса.

Фиг.9 иллюстрирует формат Пакета заголовка подкадра.

Фиг.10 иллюстрирует формат и содержимое Пакета заполнителя.

Фиг.11 иллюстрирует формат Пакета видеопотока.

Фиг.12А-12Е иллюстрируют формат и содержимое для Дескриптора формата видеоданных, используемого на фиг.11.

Фиг.13 иллюстрирует использование упакованных и неупакованных форматов для данных.

Фиг.14 иллюстрирует формат Пакета аудиопотока.

Фиг.15 иллюстрирует использование выровненных по байтам и упакованных форматов с импульсно-кодовой модуляцией (ИКМ) для данных.

Фиг.16 иллюстрирует формат Пакета определенного пользователем потока.

Фиг.17 иллюстрирует формат Пакета карты цветов.

Фиг.18 иллюстрирует формат Пакета инкапсуляции обратной линии передачи данных.

Фиг.19 иллюстрирует формат Пакета возможностей клиента.

Фиг.20 иллюстрирует формат Пакета данных клавиатуры.

Фиг.21 иллюстрирует формат Пакета данных указательного устройства.

Фиг.22 иллюстрирует формат Пакета закрытия линии передачи данных.

Фиг.23 иллюстрирует формат Пакета запроса клиента и состояния.

Фиг.24 иллюстрирует формат Пакета пересылки битовых блоков.

Фиг.25 иллюстрирует формат Пакета заливки растровой области.

Фиг.26 иллюстрирует формат Пакета заливки растровым узором.

Фиг.27 иллюстрирует формат Пакета канала данных линии обмена данными.

Фиг.28 иллюстрирует формат Пакета состояния питания дисплея.

Фиг.29 иллюстрирует формат Пакета выполнения перехода на тип.

Фиг.30 иллюстрирует формат Пакета включения аудиоканала прямой линии передачи

данных.

Фиг.31 иллюстрирует формат Пакета частоты аудиоотсчетов обратной линии передачи данных.

Фиг.32 иллюстрирует формат Пакета служебных данных защиты цифрового

5 содержимого.

Фиг.33 иллюстрирует формат Пакета разрешения прозрачного цвета.

Фиг.34 иллюстрирует формат Пакета измерения задержки на двойное прохождение.

Фиг.35 иллюстрирует временные соотношения событий во время Пакета измерения задержки на двойное прохождение.

10 Фиг.36 иллюстрирует примерную реализацию генератора ЦИК и устройства проверки, используемых для реализации изобретения.

Фиг.37A иллюстрирует временные соотношения сигналов ЦИК для устройства по фиг.36 при посылке пакетов данных.

15 Фиг.37B иллюстрирует временные соотношения сигналов ЦИК для устройства по фиг.36 при приеме пакетов данных.

Фиг.38 иллюстрирует этапы обработки для обычного запроса на обслуживание без состязания.

20 Фиг.39 иллюстрирует этапы обработки для типового запроса на обслуживание, объявленного после начала последовательности повторного запуска линии передачи данных, соперничающей с запуском линии передачи данных.

Фиг.40 иллюстрирует то, как последовательность данных может передаваться с использованием кодирования DATA-STB.

Фиг.41 иллюстрирует схему, используемую для генерирования сигналов DATA и STB из входных данных на хосте и затем восстановления данных на клиенте.

25 Фиг.42 иллюстрирует драйверы и резисторы согласованной нагрузки, используемые для реализации одного варианта осуществления.

Фиг.43a-43c иллюстрируют этапы и уровни сигналов, используемые клиентом для обеспечения обслуживания от хоста и хостом для предоставления такого обслуживания.

30 Фиг.44 иллюстрирует относительные промежутки между переходами на линии Data0, на других линиях данных (DataX) и на линиях строб-импульсов (Stb).

Фиг.45 иллюстрирует присутствие задержки в ответном сигнале, которая может иметь место, когда хост выключает драйвер хоста после пересылки пакета.

Фиг.46 иллюстрирует присутствие задержки в ответном сигнале, которая может иметь место, когда хост включает драйвер хоста для пересылки пакета.

35 Фиг.47 иллюстрирует анализ тока утечки.

Фиг.48 иллюстрирует характеристики переключения и относительные зависимости временных соотношений для времени включения и выключения выходов хоста и клиента.

40 Фиг.49 иллюстрирует высокоуровневую диаграмму этапов обработки сигналов и условия, при которых может осуществляться синхронизация с использованием конечного автомата.

Фиг.50 иллюстрирует типовые величины задержки, встречаемые для обработки сигналов по прямому и обратному трактам в системе, использующей МИЦД.

Фиг.51 иллюстрирует измерение предельной задержки на двойное прохождение.

45 Фиг.52A иллюстрирует изменения скорости передачи данных Обратной линии передачи данных.

Фиг.52B иллюстрирует пример усовершенствованного выполнения отсчетов обратных данных.

Фиг.53 иллюстрирует графическое представление значений Делителя обратной скорости передачи в зависимости от скорости передачи данных по прямой линии передачи данных.

50 Фиг.54A и 54B иллюстрируют этапы, предпринимаемые при работе интерфейса.

Фиг.55 иллюстрирует общий вид устройства интерфейса, обрабатывающего пакеты.

Фиг.56 иллюстрирует формат Пакета прямой линии передачи данных.

Фиг.57 иллюстрирует типовые значения для задержки на распространение и перекося в

интерфейсе Линии передачи данных Типа 1.

Фиг.58 иллюстрирует Data, Stb и Временные соотношения восстановления Clock (тактовых импульсов) на Линии передачи данных Типа 1 для примерной обработки сигнала при помощи интерфейса.

5 Фиг.59 иллюстрирует типовые значения для задержки на распространение и перекося в интерфейсах Линии передачи данных Типа 2, Типа 3 или Типа 4.

Фиг.60A, 60B и 60C иллюстрируют различные возможности для временных соотношений двух сигналов данных и MDDI_Stb относительно друг друга, равные идеальным, ранним и поздним соответственно.

10 Фиг.61 иллюстрирует назначение выводов интерфейса примерных соединителей, используемые с интерфейсами Типа-1/Типа 2.

Фиг.62A и 62B иллюстрируют возможные формы сигнала MDDI_Data и MDDI_Stb для обоих интерфейсов Типа 1 и Типа 2 соответственно.

15 Фиг.63 иллюстрирует высокоуровневую диаграмму альтернативных этапов обработки сигнала и условий, при которых может осуществляться синхронизация с использованием конечного автомата.

Фиг.64 иллюстрирует примерные относительные временные соотношения между последовательностью циклов тактовых импульсов и временными соотношениями различных пакетов обратной линии передачи данных и значениями делителей.

20 Фиг.65 иллюстрирует примерную обработку пересылки кода ошибки.

Фиг.66 иллюстрирует устройство, используемое для обработки пересылки кода ошибки.

Фиг.67A иллюстрирует обработку пересылки кода ошибки для перегрузки кода.

Фиг.67B иллюстрирует обработку пересылки кода ошибки для приема кода.

Фиг.68A иллюстрирует этапы обработки для иницируемого хостом пробуждения.

25 Фиг.68B иллюстрирует этапы обработки для иницируемого клиентом пробуждения.

Фиг.68C иллюстрирует этапы обработки для иницируемого хостом и клиентом пробуждения с состоянием.

Фиг.69 иллюстрирует формат Пакета запроса свойств виртуальной панели управления (ВПУ).

30 Фиг.70 иллюстрирует формат Пакета ответа свойств ВПУ.

Фиг.71 иллюстрирует формат Списка ответа свойств ВПУ.

Фиг.72 иллюстрирует формат Пакета установки свойств ВПУ.

Фиг.73 иллюстрирует формат Пакета запроса действительных параметров.

Фиг.74 иллюстрирует формат Пакета ответа действительных параметров.

35 Фиг.75 иллюстрирует формат Пакета возможностей масштабируемого видеопотока.

Фиг.76 иллюстрирует формат Пакета установки масштабируемого видеопотока.

Фиг.77 иллюстрирует формат Пакета подтверждения масштабируемого видеопотока.

Фиг.78 иллюстрирует формат Пакета масштабируемого видеопотока.

Фиг.79 иллюстрирует формат Пакета запроса заданного состояния.

40 Фиг.80 иллюстрирует формат Пакета списка ответа действительных состояний.

Фиг.81A иллюстрирует формат Пакета параметров задержки на обработку пакета.

Фиг.81B иллюстрирует формат элемента Списка параметров задержки.

Фиг.82 иллюстрирует формат Пакета возможностей персонального дисплея.

Фиг.83 иллюстрирует элементы Списка точек кривизны поля.

45 Фиг.84A иллюстрирует формат Пакета отчета об ошибках клиента.

Фиг.84B иллюстрирует формат элемента Списка отчета об ошибках.

Фиг.85 иллюстрирует формат Пакета идентификации клиента.

Фиг.86 иллюстрирует формат Пакета возможностей альтернативного дисплея.

Фиг.87 иллюстрирует формат Пакета доступа к регистрам.

50 Фиг.88A-88C иллюстрируют использование двух буферов дисплея для уменьшения видимых артефактов.

Фиг.89 иллюстрирует два буфера с обновлением изображения дисплея более быстрым, чем пересылка изображения.

Фиг.90 иллюстрирует два буфера с обновлением изображения дисплея более медленным, чем пересылка изображения.

Фиг.91 иллюстрирует два буфера с обновлением изображения дисплея значительно быстрее, чем пересылка изображения.

5 Фиг.92 иллюстрирует три буфера с обновлением изображения дисплея более быстрым, чем пересылка изображения.

Фиг.93 иллюстрирует три буфера с обновлением изображения дисплея более медленным, чем пересылка изображения.

10 Фиг.94 иллюстрирует один буфер с обновлением изображения дисплея более быстрым, чем пересылка изображения.

Фиг.95 иллюстрирует подключение хоста и клиента при помощи последовательного подключения и концентратора.

Фиг.96 иллюстрирует устройства клиента, подключенные при помощи комбинации концентраторов и последовательного подключения.

15 Фиг.97 иллюстрирует карту цветов.

Подробное описание

I. Обзор

Основной целью изобретения является создание цифрового интерфейса для мобильных дисплеев (ЦИМД), как описано ниже, который дает в результате или обеспечивает
20 экономичный механизм пересылки с малой потребляемой мощностью, который дает возможность выполнять высокоскоростную или очень высокоскоростную пересылку данных по линии обмена данными с малым радиусом действия между устройством хоста и устройством клиента, таким как элемент дисплея, используя линию или канал передачи данных «последовательного» типа. Этот механизм применяется для реализации с
25 миниатюрными соединителями и тонкими гибкими кабелями, которые особенно полезны при подключении внутреннего (внутреннего для корпуса или несущего каркаса) дисплея, или элементов или устройств вывода, или устройств ввода к центральному контроллеру или элементу или устройству обмена данными. Кроме того, этот механизм подключения очень полезен для подключения внешних элементов или устройств дисплея, таких как
30 носимые микродисплеи (дисплейные очки или проекторы), или других типов устройств представления визуальной, слышимой, тактильной информации к портативным компьютерам, беспроводным устройствам обмена данными или устройствам развлечения.

Хотя термины Мобильный и Дисплей ассоциируются с названием протокола, необходимо понять, что это только для удобства в том смысле, что если имеется
35 стандартное имя, то его легко поймет специалист в данной области техники, работающий с интерфейсом и протоколом, так как он относится к стандарту Ассоциации по стандартам в области видеоэлектроники (АСВЭ) и различным применениям этого стандарта. Однако легко понять после анализа вариантов осуществления, представленных ниже, что многие относящиеся к немобильным и недисплейным применения получают пользу от применения
40 этого протокола, приводя к структуре интерфейса или механизму пересылки, и подразумевается, что ярлык ЦИМД не означает какие-либо ограничения на сущность или полезность изобретения или его разнообразных вариантов осуществления.

Преимущество вариантов осуществления изобретения заключается в том, что предлагается метод для пересылки данных, который имеет малую сложность, малую
45 стоимость, имеет высокую надежность, хорошо подходит к сфере использования и очень устойчивый к ошибкам, в то же время оставаясь очень гибким.

Варианты осуществления изобретения могут использоваться в различных ситуациях для обмена или пересылки больших количеств данных в основном для аудио, видео или мультимедийных применений от устройства хоста или источника, где такие данные
50 генерируются, манипулируются, например, для пересылки на заданные устройства, или обрабатываются иным образом, или хранятся; на устройство клиента или приемное устройство, такое как видеодисплей или проекционный элемент, звуковые громкоговорители или другое устройство представления с высокой скоростью передачи.

Типовым применением, которое обсуждается ниже, является пересылка данных от или портативного компьютера, или беспроводного телефона или модема на устройство визуального отображения, такое как малый видеозэкран или носимый прибор с микродисплеем, такой как в виде дисплейных очков или шлемов, содержащих малые

5 проекционные объективы и экраны, или от устройства хоста на устройство клиента внутри таких компонентов, т.е. от процессора или контроллера на внутренний экран или другой элемент представления, а также от различных внутренних или внешних устройств ввода, применяющих клиента, на расположенный внутри (расположенный внутри этого же корпуса устройства или несущего каркаса) хост, или подключенный к нему кабелем или

10 проводниками.

Характеристики или свойства ЦИМД такие, что они не зависят от конкретного дисплея или технологии представления. Он представляет собой очень гибкий механизм для

15 пересылки данных с высокой скоростью, не принимая во внимание ни внутреннюю структуру этих данных, ни функциональные аспекты данных или команд, которые их реализуют. Это дает возможность корректировать временные соотношения пересылаемых

пакетов данных для приспособления к индивидуальным отличительным свойствам конкретных устройств клиента, например уникальные дисплеи требуют определенные устройства, или чтобы удовлетворить требованиям объединенного аудио и видео для

20 некоторых аудио-видео (А-В) систем, или для некоторых устройств ввода, таких как джойстики, сенсорные панели и т.п. Интерфейс является очень агностическим для элемента дисплея или устройства клиента до тех пор, пока придерживаются выбранному протоколу. Кроме того, составные данные последовательной линии передачи данных или скорость передачи данных, могут изменяться на несколько порядков по величине, что

25 дает возможность разработчику системы обмена данными или устройства хоста оптимизировать стоимость, требования к мощности, сложность устройства клиента и частоты обновления устройства клиента.

Интерфейс данных представляется, главным образом, для использования при пересылке больших количеств высокоскоростных данных по «проводной» сигнальной линии передачи данных или небольшому кабелю. Однако некоторые применения также

30 могут воспользоваться преимуществом беспроводной линии передачи данных, включая оптические линии передачи данных, при условии, что она конфигурируется для

использования одинаковых структур пакетов и данных, разработанных для протокола интерфейса, и могут поддерживать требуемый уровень пересылки с достаточно низким потреблением мощности или сложности, оставаясь целесообразным.

35 **II. Среда**

Типовое применение можно видеть на фиг.1А и 1В, где портативный компьютер или носимый компьютер 100 и беспроводный телефон или устройство 102 ПЦП показаны

40 обменивающимися данными с устройствами 104 и 106 дисплея соответственно вместе с системами 108 и 112 воспроизведения звука. Кроме того, фиг.1А изображает потенциальные подключения к большему дисплею или экрану 114 или проектору 116 изображений, которые показаны только на одной фигуре для ясности, но также могут

подключаться к беспроводному устройству 102. Беспроводное устройство может в

настоящий момент принимать данные или хранило ранее некоторое количество данных

мультимедийного типа в элементе или устройстве памяти для более позднего

45 представления для просмотра и/или прослушивания конечным пользователем беспроводного устройства. Так как типовое беспроводное устройство используется

большую часть времени для передачи речи и простого текста, оно имеет довольно

маленький экран дисплея и простую звуковую систему (громкоговорители) для передачи

информации пользователю устройства 102.

50 Компьютер 100 имеет значительно больший экран, но все же неадекватную внешнюю звуковую систему, и ему все же не хватает других устройств для представления

мультимедиа, таких как телевидение высокой четкости или киноэкраны. Компьютер 100

используется с целью иллюстрации, и другие типы процессоров, интерактивные видеоигры

или бытовые электронные приборы также могут использоваться с изобретением. Компьютер 100 может применять, но не ограничивается ими, беспроводный модем или другое встроенное устройство для беспроводного обмена данными, или подключаться к таким устройствам, используя кабель или беспроводную линию передачи данных, как требуется.

Это делает представление более сложных или «обогащенных» данных менее полезным или доставляющим удовольствие впечатлением. Поэтому промышленность разрабатывает другие механизмы и устройства для представления информации конечным пользователям и представления минимального уровня требуемого удовольствия или положительных впечатлений.

Как ранее описано выше, устройства дисплея нескольких типов были разработаны или разрабатываются в настоящее время для представления информации конечным пользователям устройства 100. Например, одна или несколько компаний разработали комплекты носимых дисплейных очков, которые проецируют изображение перед глазами пользователя устройства для представления визуального дисплея. При правильном расположении такие устройства эффективно «проецируют» виртуальное изображение, воспринимаемое глазами пользователя, которое значительно больше, чем элемент, обеспечивающий визуальный выходной результат. Т.е. очень маленький проекционный элемент дает возможность глазу (глазам) пользователя «видеть» изображения в значительно большем масштабе, чем возможно с типовыми экранами жидкокристаллических дисплеев (ЖКД) и т.п. Использование изображений на больших виртуальных экранах также дает возможность использовать изображения со значительно более высокими разрешениями, чем возможно с более ограниченными дисплеями на экранах ЖКД. Другие устройства дисплеев могут включать в себя, но не ограничиваются ими, маленькие экраны ЖКД или различные элементы с дисплеями на плоских панелях, проекционные объективы и драйверы дисплеев для проецирования изображений на поверхность и т.д.

Также могут быть дополнительные элементы, подключенные к или связанные с использованием беспроводного устройства 102 или компьютера 100, для представления выходного результата другому пользователю или другому устройству, которое, в свою очередь, передает сигналы куда-то еще или запоминает их. Например, данные могут запоминаться в флэш-памяти, в оптическом виде, например, используя записываемые носители компакт-дисков, или на магнитных носителях, таких как устройство записи на магнитную ленту и подобные устройства, для последующего использования.

Кроме того, многие беспроводные устройства и компьютеры теперь имеют встроенные возможности декодирования музыки MP3, а также другие усовершенствованные системы и декодеры звука. Портативные компьютеры в основном используют возможности воспроизведения компакт-дисков и ЦМД, и некоторые имеют маленькие специализированные считыватели флэш-памяти для приема предварительно записанных звуковых файлов. Проблема обладания такими возможностями заключается в том, что файлы цифровой музыки обещают обогащенные впечатления со значительно улучшенными свойствами, но только если не отстают характеристики процесса декодирования и воспроизведения. Это же верно для файлов цифрового видео.

Чтобы способствовать воспроизведению звука, на фиг.1А показаны внешние громкоговорители 114, которые также могут сопровождаться дополнительными элементами, такими как сабвуферы или громкоговорители «объемного звучания» для проекции переднего и заднего звука. Одновременно громкоговорители или наушники 108 указываются в виде встроенных в несущий каркас или механизм устройства 106 микродисплея на фиг.1В. Как известно, могут использоваться другие элементы воспроизведения аудио или звука, включая устройства усиления мощности или формирования звука.

В любом случае, как описано выше, когда требуется переслать данные изображения высокого качества или с высоким разрешением и высококачественную аудиоинформацию

или сигналы данных от источника данных конечному пользователю по одной или нескольким линиям 110 обмена данными, требуется высокая скорость передачи данных. Т.е. линия 110 пересылки данных безусловно представляет собой потенциальное узкое место в обмене данными, как описано ранее, и ограничивает характеристики системы, так

5 как настоящие механизмы пересылки не достигают обычно требуемых высоких скоростей передачи данных. Как описано выше, например, для более высоких разрешений изображения, таких как 1024 на 1024 пикселя, с глубиной цвета 24-32 бита на пиксель и со скоростями передачи данных 30 кадров/с, скорости передачи данных могут

10 приближаться к скоростям, превышающим 755 Мбит/с или более. Кроме того, такие изображения могут представляться как часть мультимедийного представления, которое включает в себя аудиоданные и потенциально дополнительные сигналы, имеющие дело с интерактивными играми или обменами данными или с различными командами, управлением или сигналами, дополнительно повышая количество данных и скорость передачи данных.

Также ясно, что меньшее число кабелей или межсоединений, требуемых для

15 установления линии передачи данных, означает более легкое использование мобильных устройств, связанных с дисплеем, и более вероятно, что будут приняты большим количеством пользователей. Это особенно верно там, где многочисленные устройства обычно используются для установления полного аудиовизуального впечатления, и в большей степени особенно, так как повышается уровень качества дисплеев и устройств

20 вывода звука.

Другое типовое применение, относящееся ко многим из вышеупомянутых и других улучшений в видеоэкранах и других устройствах ввода или вывода, можно видеть на фиг.1C и 1D, где портативный или носимый компьютер 130 и устройство 140 беспроводного

25 телефона или ПЦП показаны обменивающимися данными с «внутренними» устройствами 134 и 144 дисплеев соответственно вместе с системами 136 и 146 воспроизведения звука.

На фиг.2A и 2B малые вырезанные участки общих электронных устройств или продуктов используются для того, чтобы показать расположение одного или нескольких хостов и контроллеров в одной части устройства с обобщенной линией обмена данными, здесь 138 и 148 соответственно, соединяющей их с элементами или экранами видеодисплеев,

30 имеющих соответствующих клиентов, через вращающееся соединение некоторого известного типа, используемое в настоящее время в электронной промышленности. Можно видеть, что количество данных, составляющих данные пересылки, требует того, чтобы линии 138 и 148 передачи данных имели большое количество проводников. Оценивается, что такие линии обмена данными приближаются к 90 или более проводникам, чтобы

35 удовлетворять современным растущим потребностям для использования усовершенствованных цветных и графических интерфейсов, элементов дисплеев, на таких устройствах из-за типов параллельных или других известных методов интерфейса, доступных для пересылки таких данных.

К сожалению, более высокие скорости передачи данных превосходят современную

40 технологию, доступную для пересылки данных, как в смысле необработанного количества данных, необходимого для пересылки на единицу времени, так и в смысле производства надежных экономически эффективных механизмов физической пересылки.

Требуется метод, структура, средство или способ для пересылки данных с более высокими скоростями передачи для линии пересылки данных или тракта обмена данными

45 между элементами представления и источником данных, которые предусматривают соответственно малую (меньшую) мощность, малый вес и максимально простую и экономичную кабельную структуру. Заявители разработали новый метод или способ и устройство для достижения этих и других целей, чтобы дать возможность множеству мобильных, портативных или даже стационарных устройств пересылать данные на

50 требуемые дисплеи, микродисплеи или элементы пересылки аудио с очень высокими скоростями передачи данных, в то же время сохраняя требуемую малую потребляемую мощность и сложность.

III. Архитектура системы цифрового интерфейса передачи данных с высокой

скоростью

Чтобы создать и эффективно использовать новый интерфейс устройств, были сформулированы протокол передачи сигналов и архитектура системы, которые обеспечивают очень высокую скорость пересылки данных, используя сигналы малой мощности. Протокол основывается на структуре пакетов и общих кадров или структурах, связанных вместе для формирования протокола для передачи выбранного набора данных или типов данных вместе с структурой команд или рабочей структурой, налагаемой на интерфейс.

А. Обзор

Устройства, соединенные посредством или обменивающие данные по линии передачи данных ЦИМД, называются хостом и клиентом, причем клиентом обычно является устройство дисплея некоторого типа, хотя предполагаются другие устройства вывода и ввода. Данные от хоста на дисплей передаются в прямом направлении (упоминаемом как прямой трафик или линия передачи данных), и данные от клиента на хост передаются в обратном направлении (обратный трафик или линия передачи данных), разрешенном хостом. Это иллюстрируется в базовой конфигурации, показанной на фиг.3. На фиг.3 хост 202 подсоединен к клиенту 204, используя двунаправленный канал 206 обмена данными, который изображается как содержащий прямую линию 208 передачи данных и обратную линию 210 передачи данных. Однако эти каналы формируются обычным набором проводников, пересылка данных по которым эффективно переключается между операциями прямой или обратной линий передачи данных. Это предусматривает значительно уменьшенное количество проводников, непосредственно касающееся одной из многих проблем, встречаемых с настоящими подходами к высокоскоростной пересылке данных в маломощных средах, таких как для мобильных электронных устройств.

Как описано в другом месте, хост содержит одно из устройств нескольких типов, которые могут получить пользу от использования настоящего изобретения. Например, хост 202 может быть портативным компьютером в виде карманного, носимого или подобного мобильного вычислительного устройства. Он также может быть персональным цифровым помощником (ПЦП), пейджинговым устройством или одним из многих беспроводных телефонов или модемов. Альтернативно хост 202 может быть портативным развлекательным устройством или устройством представления, таким как портативный проигрыватель на ЦМД или компакт-дисках или устройством для воспроизведения игры.

Кроме того, хост может находиться в качестве устройства хоста или элемента управления в многочисленных широко используемых или планируемых коммерческих продуктах, для которых с клиентом требуется высокоскоростная линия обмена данными. Например, хост может использоваться для пересылки данных с высокими скоростями от устройства видеозаписи основанному на хранении клиенту для улучшенной характеристики или большому экрану с высоким разрешением для представления. Прибор, такой как холодильник, который включает в себя встроенную систему управления запасами или вычислительную систему и/или подключения «Bluetooth» к другим домашним устройствам, может иметь улучшенные возможности дисплея при работе в режиме подключения к интернету или сети «Bluetooth» или уменьшил потребности проводки для дисплеев в двери (клиент) и клавиатуры или сканеров (клиент), тогда как электронный компьютер или системы управления (хост) находятся где-то еще в стойке. В общем, специалист в данной области техники примет во внимание большое разнообразие современных электронных устройств и приборов, которые могут получить пользу от использования данного интерфейса, а также возможности модифицирования более старых устройств посредством передачи информации с более высокой скоростью передачи данных, использующих ограниченное количество проводников, доступных или во вновь добавляемых или существующих соединителях или кабелях.

Одновременно клиент 204 может содержать многочисленные устройства, используемые для представления информации конечному пользователю или представления информации от пользователя хосту. Например, микродисплей, встроенный в дисплейные очки или

обычные очки, проекционное устройство, встроенное в шляпу или шлем, маленький экран или даже голографический элемент, встроенный в транспортное средство, например в окно или ветровое стекло, или различные системы громкоговорителей, головных телефонов или звуковые системы для представления высококачественного звука или музыки. Другие
 5 устройства представления включают в себя проекторы или проекционные устройства, используемые для представления информации о встречах или для изображений фильмов и телевидения. Другим примером является использование сенсорных панелей или чувствительных устройств, устройств ввода на основе распознавания голоса, сканеров безопасности и т.п., которые могут быть вызваны для пересылки значительного
 10 количества информации от пользователя устройства или системы с незначительным фактическим «вводом» кроме прикосновения или звука от пользователя. Кроме того, установочные станции для компьютеров и автомобильные комплекты или настольные комплекты или держатели для беспроводных телефонов могут служить в качестве устройств интерфейсов для конечных пользователей или для других устройств и
 15 оборудования и использовать или клиенты (устройства вывода или ввода, такие как мыши), или хосты, чтобы способствовать пересылке данных особенно там, где участвуют высокоскоростные сети.

Однако специалисту в данной области техники понятно, что настоящее изобретение не ограничивается этими устройствами, причем существует много других устройств на рынке
 20 и предлагаемых для использования, которые предназначены для предоставления конечным пользователям высококачественных изображений и звука или в смысле хранения и транспортировки, или в смысле представления при воспроизведении. Настоящее изобретение полезно для повышения пропускной способности передачи данных между различными элементами или устройствами, чтобы приспособить высокие скорости
 25 передачи данных, необходимые для реализации требуемого впечатления пользователя.

Обладающие признаками изобретения ЦИМД и протокол сигналов обмена данными могут использоваться для упрощения межсоединений между процессором, контроллером хоста или схемным компонентом (например) и дисплеем внутри устройства или корпуса устройства или конструкции (упоминаемый как внутренний режим), чтобы снизить
 30 стоимость или сложность и связанные с ними требования на мощность и управление, или ограничить эти соединения и повысить надежность, не только для подключения к или для внешних элементов, устройств или оборудования (упоминаемый как внешний режим).

Скорость передачи данных составной последовательной линии передачи данных по каждой сигнальной паре, используемой данной структурой интерфейса, может изменяться
 35 на многие порядки по величине, что дает возможность разработчику системы или устройства легко оптимизировать стоимость, мощность, сложность реализации и частоту обновления изображения дисплея для данного применения или назначения. Свойства ЦИМД не зависят от технологии дисплея или другого устройства представления (целевого клиента). Временные соотношения пакетов данных, пересылаемых через интерфейс, легко
 40 могут корректироваться так, чтобы приспособляться к индивидуальным отличительным свойствам конкретных клиентов, таких как устройства дисплея, звуковые системы, память и элементы управления, или к объединенным требованиям на временные соотношения аудио-видеосистем. Хотя это делает возможным иметь систему с очень малой потребляемой мощностью, не является требованием для различных клиентов иметь
 45 буферы кадров, чтобы использовать протокол ЦИМД по меньшей мере на некотором уровне.

В. Типы интерфейсов

Интерфейс ЦИМД рассматривается как относящийся по меньшей мере к четырем и потенциально более интерфейсам в некоторой степени отдельных физических типов,
 50 встречаемых в промышленности средств связи и средств вычислительной техники. Они обозначены просто как Тип 1, Тип 2, Тип 3 и Тип 4, хотя другие метки или обозначение могут применяться специалистами в данной области техники в зависимости от конкретных применений, для которых они используются, или промышленности, с которой они связаны.

Например, простые аудиосистемы используют меньшее количество соединений, чем более сложные мультимедийные системы и могут ссылаться на свойства, такие как «каналы» различным образом, и т.п.

Интерфейс Типа 1 конфигурируется как 6-проводный интерфейс или другого типа проводников или проводящих элементов, который делает его пригодным для мобильных или беспроводных телефонов, ПЦП, электронных игр и портативных медиаплееров, таких как проигрыватели компакт-дисков, или МР3-плееры, и подобных устройств или устройств, используемых в подобных типах электронной бытовой техники. В одном варианте осуществления интерфейс может конфигурироваться как 8-проводной (проводниковый) интерфейс, который больше подходит для носимых компьютеров, ноутбуков или настольных персональных компьютеров и аналогичных устройств или применений, которые не требуют быстрых обновлений данных и которые не имеют встроенного контроллера линии передачи данных ЦИМД. Этот тип интерфейса также отличается использованием дополнительного двухпроводного интерфейса универсальной последовательной шины (УПШ), который очень полезен для обеспечения поддержки существующих операционных систем или программного обеспечения, имеющейся на большинстве персональных компьютеров.

Интерфейсы Типа 2, Типа 3 и Типа 4 пригодны для клиентов или устройств с высокими рабочими характеристиками и используют большую, более сложную кабельную систему с дополнительными проводниками типа витой пары для обеспечения соответствующего экранирования и пересылок с малыми потерями для сигналов данных.

Интерфейс Типа 1 пропускает сигналы, которые могут содержать информацию для дисплея, аудиоинформацию, управляющую информацию и ограниченную информацию сигнализации, и типично используется для мобильных клиентов или устройств клиентов, которые не требуют полноскоростных видеоданных с высоким разрешением. Интерфейс Типа 1 может легко поддерживать разрешение супервидеографической матрицы (СВГМ) с частотой 30 кадров/с плюс аудио по схеме 5.1 каналов, и в минимальной конфигурации может использовать только в целом три пары проводов, две пары для передачи данных и одну пару для подачи питания. Этот тип интерфейса, главным образом, предназначен для устройств, таких как мобильные беспроводные устройства, где хост УПШ обычно недоступен в таком устройстве для подключения и пересылки сигналов. В данной конфигурации мобильным беспроводным устройством является устройство хоста ЦИМД, и оно действует в качестве «ведущего устройства», которое управляет линией обмена данными с хоста, который в основном посылает данные клиенту (прямой трафик или линия передачи данных) для представления, отображения или воспроизведения.

В данном интерфейсе хост включает прием данных обмена на хосте с клиента (обратный трафик или линия передачи данных) посредством посылки специальной команды или типа пакета клиенту, который позволяет ему захватить шину (линию передачи данных) в течение заданной продолжительности и послать данные хосту в виде обратных пакетов. Это изображено на фиг.4, где тип пакета, упоминаемый как пакет инкапсуляции (описан ниже), используется для обеспечения пересылки обратных пакетов по линии пересылки данных, создавая обратную линию передачи данных. Временной интервал, выделяемый хосту для опрашивания клиента в отношении данных, предварительно определяется хостом и основывается на требованиях каждого конкретного применения. Этот тип полудуплексной двунаправленной пересылки данных особенно выгоден там, где недоступен порт УПШ для пересылки информации или данных с клиента.

Высококачественные дисплеи, обладающие возможностями аналогичных высоких разрешений или типа ТВЧ, требуют потоков данных со скоростью передачи примерно 1,5 Гбит/с для поддержки видео с полноценным движением. Интерфейс Типа 2 поддерживает высокие скорости передачи данных посредством параллельной передачи 2 битов, интерфейс Типа 3 - посредством параллельной передачи 4 битов и интерфейс Типа 4 передает параллельно 8 битов. Интерфейсы Типа 2 и Типа 3 используют один и тот же кабель и соединитель, что и Тип 1, но могут работать с удвоенной и учетверенной

скоростями передачи данных для поддержки применений видео с высокими рабочими характеристиками на портативных устройствах. Интерфейс Типа 4 подходит для клиентов или дисплеев с очень высокими рабочими характеристиками и требует несколько больший кабель, который содержит дополнительные сигналы данных на витой паре.

5 Протокол, используемый ЦИМД, позволяет каждому хосту Типа 1, 2, 3 или 4 в основном осуществлять обмен данными с любым клиентом Типа 1, 2, 3 или 4 посредством согласования, какая возможна самая высокая скорость передачи данных, которая может быть использована. Возможности или доступные свойства того, что может быть упомянуто как устройство с наименьшими возможностями, используются для установления рабочих характеристик линии передачи данных. Как правило, даже для систем, в которых хост и клиент оба способны использовать интерфейсы Типа 2, Типа 3 или Типа 4, оба начинают работу в качестве интерфейса Типа 1. Затем хост определяет возможности целевого клиента и согласовывает операцию перехода или повторного конфигурирования в режим или Типа 2, или Типа 3, или Типа 4, что больше подходит для конкретного применения.

15 Как правило, возможно, чтобы хост использовал надлежащий протокол канального уровня (описанный подробно ниже) и, как правило, в любой момент, переходил на шаг вниз или выполнял снова операцию повторного конфигурирования на более медленный режим для экономии мощности или переходил на шаг вверх на более быстрый режим для поддержки пересылок с более высокими скоростями, например, для содержимого дисплея с более высоким разрешением. Например, хост может изменить типы интерфейса, когда система переключается с источника питания, такого как батареи, на питание от сети переменного тока или когда источник носителя данных дисплея переключается в формат с меньшим или большим разрешением, или комбинация этих и других условий или событий может рассматриваться как основание для изменения типа интерфейса или режима пересылки.

25 Также возможно, чтобы система производила обмен данными, используя один режим в одном направлении и другой режим в другом направлении. Например, режим интерфейса Типа 4 может использоваться для пересылки данных на дисплей с высокой скоростью передачи, тогда как режим Типа 1 используется при пересылке данных на устройство хоста с периферийных устройств, таких как клавиатура или указательное устройство. Для специалиста в данной области техники понятно, что хосты и клиенты могут обмениваться исходящими данными на различных скоростях.

30 Часто, пользователи протокола ЦИМД могут отличать «внешний» режим от «внутреннего» режима. Внешний режим описывает использование протокола и интерфейса для подключения хоста в одном устройстве к клиенту вне этого устройства, которое находится примерно до 2 метров или около этого от хоста. В данной ситуации хост также может подавать питание внешнему клиенту, так что оба устройства легко могут работать в мобильной среде. Внутренний режим описывает то, что хост подключается к клиенту, содержащемуся внутри этого же устройства, например внутри общего корпуса или несущего каркаса или конструкции некоторого вида. Примером являются применения внутри беспроводного телефона или другого беспроводного устройства, или портативного компьютера, или игрового устройства, где клиентом является дисплей или драйвер дисплея, или устройства ввода, такого как клавиатура или сенсорная панель, или звуковая система, и хост представляет собой центральный контроллер, графическую машину или элемент центрального процессора (ЦП). Так как клиент располагается гораздо ближе к хосту в применениях внутреннего режима в противоположность применениям внешнего режима, как правило, нет требований, описанных для подачи питания клиенту в таких конфигурациях.

С. Физическая структура интерфейса

50 Общее расположение контроллера устройства или линии передачи данных для установления обмена данными между устройствами хоста и клиента изображено на фиг.5 и 6. На фиг.5 и 6 контроллер 402 и 502 линии передачи данных ЦИМД показан установленным в устройстве 202 хоста, и контроллер 404 и 504 линии передачи данных

ЦИМД показан установленным в устройстве 204 клиента. Как и раньше, хост 202 подключается к клиенту 204 с использованием двунаправленного канала 406 обмена данными, содержащего группу проводников. Как описано ниже, контроллеры линии передачи данных как хоста, так и клиента могут быть изготовлены в виде интегральной

5 схемы, использующей одно схемное решение, которая может быть установлена, отрегулирована или запрограммирована для работы в качестве или контроллера (драйвера) хоста, или контроллера (приемника) клиента. Это обеспечивает малую стоимость вследствие более крупномасштабного производства одного схемного устройства.

10 На фиг.6 контроллер 502 линии передачи данных ЦИМД показан установленным в устройстве 202' хоста, и контроллер 504 линии передачи данных ЦИМД показан установленным в устройстве 204' клиента. Как и раньше, хост 202' подключается к клиенту 204' с использованием двунаправленного канала 506 обмена данными, содержащего группу проводников. Как описано выше, контроллеры линии передачи данных

15 как хоста, так и клиента могут быть изготовлены с использованием одного схемного решения.

На фиг.5 и 6 также изображены сигналы, передаваемые между хостом и клиентом, таким как устройство дисплея, по линии передачи данных ЦИМД или используемым физическим проводникам. Как показано на фиг.5 и 6, первичный тракт или механизм для пересылки

20 данных через ЦИМД использует сигналы данных, обозначенные как MDDI_Data0+/- и MDDI_Stb+/- . Каждый из них представляет собой сигнал данных низкого напряжения, который пересылается по дифференциальной паре проводов в кабеле. Существует только один переход или на паре MDDI_Data0, или на паре MDDI_Stb для каждого бита, посылаемого по интерфейсу. Он представляет собой механизм пересылки, основанный на

25 напряжении, а не основанный на токе, так что статическое потребление тока равно почти нулю. Хост возбуждает сигналы MDDI_Stb для дисплея клиента.

Хотя данные могут протекать как в прямом, так и в обратном направлении по парам MDDI_Data, т.е. они представляют собой двунаправленный тракт пересылки, хост является ведущим устройством или контроллером линии передачи данных. Тракты сигналов

30 MDDI_Data0 и MDDI_Stb работают в дифференциальном режиме, чтобы максимизировать помехоустойчивость. Скорость передачи данных для сигналов на этих линиях определяется частотой тактовых импульсов, посылаемых хостом и изменяется в диапазоне от примерно 1 кбит/с до 400 Мбит/с или более.

Интерфейс Типа 2 содержит одну дополнительную пару проводников или трактов

35 передачи данных кроме той, которая находится в интерфейсе Типа 1, упоминаемую как MDDI_Data1+/- . Интерфейс Типа 3 содержит две дополнительные пары передачи данных или тракты сигналов кроме тех, которые находятся в интерфейсе Типа 2, упоминаемые как MDDI_Data2+/- и MDDI_Data3+/- . Интерфейс Типа 4 содержит еще четыре пары передачи данных или тракта сигналов кроме тех, которые находятся в интерфейсе Типа 3,

40 упоминаемые как: MDDI_Data4+/-, MDDI_Data5+/-, MDDI_Data6+/- и MDDI_Data7+/- соответственно. В каждой из вышеупомянутых конфигураций интерфейса хост может подавать питание клиенту или дисплею, используя проводную пару или сигналы, обозначенные как HOST_Pwr и HOST_Gnd (заземление). Как дополнительно описано ниже, также может быть предусмотрена подача питания, если требуется, в некоторых

45 конфигурациях по проводникам MDDI_Data4+/-, MDDI_Data5+/-, MDDI_Data6+/- или MDDI_Data7+/-, когда используется интерфейс «Тип», который применяет меньшее количество проводников, чем доступно или присутствует для других режимов. Эта Подача питания в основном используется для внешних режимов, там в основном нет необходимости во внутренних режимах, хотя некоторые применения могут отличаться.

50 Сводка сигналов, передаваемых между хостом и клиентом (дисплеем) по линии передачи данных ЦИМД для различных режимов, приведена ниже в таблице I в соответствии с типом интерфейса.

Таблица I			
Тип 1	Тип 2	Тип 3	Тип 4

HOST_PwrGnd	HOST_PwrGnd	HOST_PwrGnd	HOST_PwrGnd
MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-
MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-
	MDDI_Data1+/-	MDDI_Data1+/-	MDDI_Data1+/-
		MDDI_Data2+/-	MDDI_Data2+/-
		MDDI_Data3+/-	MDDI_Data3+/-
Дополнит. пит.	Дополнит. пит.	Дополнит. пит.	MDDI_Data4+/-
Дополнит. пит.	Дополнит. пит.	Дополнит. пит.	MDDI_Data5+/-
Дополнит. пит.	Дополнит. пит.	Дополнит. пит.	MDDI_Data6+/-
Дополнит. пит.	Дополнит. пит.	Дополнит. пит.	MDDI_Data7+/-

Также отметьте, что соединения HOST_Pwr/Gnd для пересылки с хоста предусматриваются в основном для внешних режимов. Внутренние применения или режимы работы в основном имеют клиентов, которые отбирают питание непосредственно от других внутренних ресурсов и не используют ЦИМД для управления распределением питания, что является очевидным для специалиста в данной области техники, так что такое распределение здесь более подробно не описывается. Однако, конечно, возможно выполнять распределение питания через ЦИМД, чтобы предусмотреть некоторые виды удобства управления питанием, синхронизации или межсоединений, например, что понятно для специалиста в данной области техники.

Кабели, используемые, как правило, для осуществления вышеуказанной конструкции и принципа действия, номинально имеют длину порядка 1,5 м, в основном 2 метра или меньше, и содержат три витые пары проводников, каждый из которых, в свою очередь, представляет собой многожильный провод 30 по американскому сортаменту проводов (АСП). Экранное покрытие из фольги оборачивается или формируется иным образом, вокруг трех витых пар в качестве дополнительного дренажного провода. Витые пары и дренажный проводник экрана заканчиваются в соединителе клиента, при этом экран подсоединяется к экрану клиента, и имеется изолирующий слой, покрывающий весь кабель, что хорошо известно в этой области техники. Провода разделены на пары: HOST_Gnd с HOST_Pwr; MDDI_Stb+ с MDDI_Stb-; MDDI_Data0+ с MDDI_Data0-; MDDI_Data1+ с MDDI_Data1- и т.д. Однако могут использоваться многочисленные проводники и кабельные системы, что известно в данной области техники, для реализации вариантов осуществления изобретения в зависимости от заданных применений. Например, более тяжелые внешние покрытия или даже металлические слои могут использоваться для защиты кабеля в некоторых применениях, в то же время более тонкие, более плоские проводящие конструкции ленточного типа могут хорошо подходить для других применений.

D. Типы данных и скорости передачи данных

Для того чтобы получить полезный интерфейс для полного спектра впечатлений пользователя и применений, мобильный интерфейс цифровых данных (МИЦД) обеспечивает поддержку многочисленной информации для клиентов и дисплеев, преобразователей звука, клавиатур, указательных устройств и многих других устройств ввода или вывода, которые могут быть встроены в или работать во взаимодействии с мобильным устройством обмена данными, вычислительным устройством или устройством дисплея вместе с информацией управления и их комбинацией. ЦИМД предназначен для того, чтобы он мог обеспечивать потоки данных многочисленных потенциальных типов, передающихся между хостом и клиентом по направлениям или прямой, или обратной линии передачи данных, используя минимальное количество кабелей или проводников. Поддерживаются как изохронные потоки, так и асинхронные потоки (обновления). Возможны многочисленные комбинации типов данных, пока скорость передачи составных данных меньше или равна максимально требуемой скорости линии передачи данных ЦИМД, которая ограничивается максимальной последовательной скоростью и используемым количеством пар данных. Они могут включать в себя, но не ограничиваются ими, те элементы, которые перечислены ниже в таблицах II и III.

	Таблица II
Пересылка с Хоста на Клиент	

изохронные видеоданные	720×480, 12 бит, 30 кадр/с	~124,5 Мбит/с
изохронные стереофонические аудиоданные	44,1 кГц, 16 бит, стерео	~1,4 Мбит/с
асинхронные графические данные	800×600, 12 бит, 10 кадр/с, стерео	~115,2 Мбит/с
асинхронное управление	Минимум	<<1,0 Мбит/с

5

Таблица III		
Пересылка с Клиента на Хост		
изохронные речевые данные	8 кГц, 8 бит	<<1,0 Мбит/с
изохронные видеоданные	640×480, 12 бит, 24 кадр/с	~88,5 Мбит/с
асинхронное состояние, ввод пользователем и т.д.	Минимум	<<1,0 Мбит/с

10

Интерфейс не является неизменным, но является расширяемым, так что он может поддерживать пересылку многочисленных «типов» информации, которые включают в себя определяемые пользователем данные, для будущей гибкости системы. Конкретными примерами данных, которые он должен обеспечивать, являются: видео с полноценным движением или в виде растровых полей с полным или частичным экраном, или сжатое видео; неподвижные растровые изображения при низкой скорости передачи для экономии мощности и снижения затрат на реализацию; ИКМ- или сжатые аудиоданные с различными разрешениями или скоростями передачи; положение и выбор указательного устройства и определяемые пользователем данные для возможностей, которые еще должны быть определены. Такие данные также могут пересылаться вместе с информацией об управлении или состоянии для обнаружения возможностей устройства или установки рабочих параметров.

15

20

25

30

35

Варианты осуществления изобретения создают прогресс в области техники для использования при пересылке данных, которые включает в себя, но не ограничивается ими: просмотр фильмов (отображение видео и аудио); использование персонального компьютера с ограниченным персональным просмотром (отображение графики, иногда объединенное с видео и аудио); воспроизведение видеоигры на персональном компьютере (ПК), консоли или персональном устройстве (отображение графики с движением, или искусственные видео и аудио); «серфинг» по интернету, используя устройства в виде видеотелефона (двунаправленное низкоскоростное видео и аудио), камеру для неподвижных цифровых фотографий, или записывающую видеокамеру для захвата цифровых видеоизображений; использование телефона, компьютера или ПЦП, установленного в проекторе, для получения представления, или установленного в настольную установочную станцию, подключенную к видеомонитору, клавиатуре и мыши; и для повышения эффективности или развлекательного использования при помощи сотовых телефонов, интеллектуальных телефонов или ПЦП, включая беспроводные указательные устройства и данные клавиатуры.

40

Интерфейс высокоскоростных данных, как описано ниже, представлен на основе предоставления больших количеств данных А-В типа по линии обмена или пересылки данных, которая обычно конфигурируется как проводная линия или линия передачи данных кабельного типа. Однако очевидно, что структура, протоколы, временные соотношения или механизм пересылки сигналов могут регулироваться для получения линии передачи данных в виде оптической или беспроводной среды, если она может поддерживать требуемый уровень пересылки данных.

45

50

Сигналы ЦИМД используют принцип, известный как частота общих кадров (ЧОК) для базового протокола или структуры сигналов. Идея, лежащая в основе использования частоты общих кадров, заключается в создании импульса синхронизации для одновременных изохронных потоков данных посредством посылки Пакетов заголовка подкадра с частотой, которая может использоваться для извлечения тактовых импульсов или временных соотношений кадра для многочисленных потоков. Частота, с которой посылаются Пакеты заголовка подкадра, представляет собой Частоту общих кадров. Устройство клиента может использовать эту частоту общих кадров в качестве временной ссылки. Низкая ЧОК повышает эффективность канала посредством уменьшения количества служебных сигналов для передачи заголовка подкадра. С другой стороны,

высокая ЧОК снижает время ожидания и допускает меньший буфер данных регулируемой емкости для аудиоотсчетов. ЧОК интерфейса настоящего изобретения динамически программируется и может устанавливаться на одно из многих значений, которые подходят для изохронных потоков, используемых в конкретном применении. Т.е. величина ЧОК

5 выбирается такой, чтобы наилучшим образом подходить к данной конфигурации клиента и хоста, как требуется.

Количество байтов, необходимое, как правило, на подкадр, которое подстраивается или программируется, для изохронных потоков данных, которые, как наиболее вероятно, должны использоваться в применении, таком как для видео или микродисплея, показано в

10 таблице IV.

Таблица IV							
Частота общих кадров (ЧОК)=300 Гц							
	X	Y	Бит	Частота кадров	Канал	Скорость (Мбит/с)	Байт/ подкадр
Компьютерная игра	720	480	24	30	1	248,832	103680
Компьютерная графика	800	600	24	10	1	115,200	48000
Видео	640	480	12	29,97 или 30	1	221,184	92160
Звуковой компакт-диск	1	1	16	44100	2	1,4112	588
Речь	1	1	8	8000	1	0,064	26-2/3

Дробное число байтов на подкадр легко получается, используя простую программируемую конструкцию счетчика M/N. Например, число 26-2/3 байтов на подкадр осуществляется посредством пересылки 2 подкадров, содержащих 27 байтов каждый, за которым следует один подкадр, содержащий 26 байтов. Меньшие ЧОК могут выбираться для получения целого числа байтов на подкадр. Однако, вообще говоря, для реализации простого счетчика M/N аппаратными средствами потребовалось бы меньше площади на кристалле интегральной схемы или электронного модуля, используемых для реализации

20 части или всех вариантов осуществления изобретения, чем площадь, необходимая для буфера обратного магазинного типа с большим количеством аудиоотсчетов.

Примерным применением, которое иллюстрирует использование различных скоростей пересылки данных и типов данных, является система караоке. Что касается караоке, это система, где конечный пользователь или пользователи поет вместе с музыкальной

30 видеопрограммой. Слова песни отображаются где-то обычно в нижней части экрана, так что пользователь знает слова, которые нужно петь, и грубо темп песни. Для этого применения требуется видеодисплей с нечастыми обновлениями графики и смешивание голоса, или голосов, пользователя со стереофоническим аудиопотоком.

Если частоту общих кадров принимают равной 300 Гц, то тогда каждый подкадр будет состоять из: 92160 байтов содержимого видео и 588 байтов содержимого аудио (основываясь на 147 16-битовых отсчетах, в стерео) на прямой линии передачи данных к клиенту, и в среднем 29,67 (26-2/3) байтов голоса посылаются обратно с микрофона на мобильное устройство караоке. Асинхронные пакеты пересылаются между хостом и

40 клиентом, возможно, установленном на голове дисплея. Они включают в себя нижнюю высоту в четверть экрана, обновляемую текстом песни на 1/30 части секундных интервалов или периодов, и другие разнообразные команды управления и состояния, посылаемые в подкадрах, когда текст песни не обновляется.

В таблице V показано, как данные распределяются в подкадре для примера с караоке. Полная используемая скорость передачи выбирается равной примерно 279 Мбит/с. Несколько более высокая скорость 280 Мбит/с позволяет пересылать примерны другие 400 байтов данных на подкадр, что позволяет использовать случайные сообщения управления и состояния.

Таблица V		
Скорость передачи элемента	Служебные байты на подкадр	Медиа байты на подкадр
Музыкальное видео при 640*480 пикселях и 30 кадр/с	2.28=56	92160
Текст слов песни при 640*120 пикселях и 1 кадр/с, обновляемый через 10 подкадров, 1/30 с.	28	23040
Звуковой компакт-диск при 44100 отсчет/с, стерео, 16 бит	2.16=32	588

Голос при 8000 отсчет/с, моно, 8 бит	28+8+8+(4.16)+(3.27)= 125	26,67 (27 максимум)
Заголовок подкадра	22	
Общее количество байт/ОК (общий кадр)	263	115815
Суммарная скорость передачи (Мбит/с)	(263+115815)·8·300=278,5872	

5 III. (Продолжение) Архитектура системы интерфейса цифровых данных с высокой скоростью передачи

Е. Канальный уровень

Данные, пересылаемые с использованием сигналов высокоскоростных последовательных данных ЦИМД, состоят из потока мультиплексируемых во времени пакетов, которые связаны друг за другом. Даже когда у передающего устройства нет данных для пересылки, контроллер линии передачи данных ЦИМД, как правило, автоматически посылает пакеты заполнителя, таким образом сохраняя поток пакетов. Использование простой структуры пакетов обеспечивает надежные изохронные временные соотношения для сигналов видео и аудио или потоков данных.

Группы пакетов заключаются в сигнальные элементы или структуры, упоминаемые как подкадры, и группы подкадров заключаются в сигнальные элементы или структуры, упоминаемые как медиакадры. Подкадр содержит один или несколько пакетов в зависимости от использования их соответствующего размера и пересылки данных, и медиакадр содержит один или несколько подкадров. Самый большой подкадр, предусматриваемый протоколом, используемым вариантами осуществления, представленными здесь, составляет порядка $2^{32}-1$, или 4294967295 байтов, и размер самого большого медиакадра тогда становится равным порядка $2^{16}-1$, или 65535 подкадров.

Как описано ниже, специальный пакет заголовка подкадра содержит уникальный идентификатор, который находится в начале каждого подкадра. Этот идентификатор также используется для получения согласования во времени кадра на устройстве клиента, когда инициируется обмен данными между хостом и клиентом. Достижение согласования во времени линии передачи данных описывается более подробно ниже.

Обычно изображение на экране дисплея обновляется один раз за медиакадр, когда отображается видео с полноценным движением. Частота кадров дисплея одинакова с частотой медиакадров. Протокол линии передачи данных поддерживает видео с полноценным движением по всему дисплею или только по небольшой области содержимого видео с полноценным движением, окруженного неподвижным изображением, в зависимости от требуемого применения. В некоторых маломощных мобильных применениях, таких как просмотр веб-страниц или электронной почты, может потребоваться только случайное обновление изображения экрана дисплея. В этих случаях выгодно передавать один подкадр, а затем отключать или инактивировать линию передачи данных для минимизации потребляемой мощности. Интерфейс также поддерживает эффекты, такие как стереоскопическое зрение, и обрабатывает графические примитивы.

Подкадры дают возможность системе включать передачу высокоприоритетных пакетов на периодической основе. Это делает возможным совместное одновременное существование изохронных потоков с минимальной величиной буферизации данных. Это представляет собой одно преимущество, которое варианты осуществления обеспечивают для процесса отображения, позволяя многочисленным потокам данных (высокоскоростной обмен видео, голоса, управления, состояния, данных указательного устройства и т.д.) по существу совместно использовать общий канал. Он пересылает информацию, используя относительно небольшое количество сигналов. Он также дает возможность существовать действиям, характерным для технологии отображения, таким как горизонтальные синхрои импульсы и интервалы гашения для монитора на электронно-лучевой трубке (ЭЛТ), или для других действий, характерных для технологии клиента.

Ф. Контроллер линии передачи данных

Контроллер линии передачи данных ЦИМД, показанный на фиг.5 и 6, изготавливается или собирается так, что он является полностью цифровой реализацией за исключением

дифференциальных линейных приемников, которые используются для приема сигналов данных и строб-сигналов ЦИМД. Однако даже дифференциальные линейные драйверы и приемники могут быть реализованы такими же цифровыми интегральными схемами, что и контроллер линии передачи данных, такими как при производстве интегральных схем (ИС) типа КМОП. Не требуются аналоговые функции или системы фазовой автоподстройки частоты (ФАПЧ) для восстановления битов или для реализации аппаратных средств для контроллера линии передачи данных. Контроллеры линии передачи данных хоста и клиента содержат очень подобные функции за исключением интерфейса клиента, который содержит конечный автомат для синхронизации линии передачи данных. Поэтому варианты осуществления изобретения предоставляют практическое преимущество возможности создания одной конструкции или схемы контроллера, которая может конфигурироваться или как хост, или как клиент, что в целом может снизить производственные затраты на контроллеры линии передачи данных.

IV. Протокол линии передачи данных интерфейса

A. Структура кадра

На фиг.7 изображен протокол передачи сигналов или структура кадра, используемая для осуществления обмена данными по прямой линии передачи данных для пересылки пакетов. Как показано на фиг.7, информация или цифровые данные группируются в элементы, известные как пакеты. Многочисленные пакеты в свою очередь группируются вместе и формируют то, что упоминается как «подкадр», и многочисленные подкадры в свою очередь группируются вместе и формируют «медиакادر». Для управления формированием кадров и пересылкой подкадров каждый подкадр начинается со специально предварительно определенного пакета, упоминаемого как Пакет заголовка подкадра (ПЗП).

Устройство хоста выбирает скорость передачи данных, подлежащую использованию для данной пересылки. Эта скорость передачи может динамически изменяться устройством хоста, основываясь как на максимальных возможностях по пересылке хоста, или данных, извлекаемых из источника хостом, так и на максимальных возможностях клиента, или другого устройства, на которое пересылаются данные.

Устройство клиента получателя, предназначенное для работы с ЦИМД или протоколом передачи сигналов изобретения или способное работать с ними, может быть опрошено хостом для определения максимальной или максимальной в настоящее время скорости пересылки данных, которую оно может использовать, или может быть использована меньшая минимальная скорость передачи по умолчанию, а также используемых типов данных и поддерживаемых свойств. Эта информация может пересылаться с использованием Пакета возможностей клиента (ПВК), как описано подробно ниже. Устройство дисплея клиента способно пересылать данные или производить обмен данными с другими устройствами, используя интерфейс с предварительно выбранной минимальной скоростью передачи данных или в пределах диапазона минимальных скоростей передачи данных, и хост выполняет запрос, используя скорость передачи данных в пределах этого диапазона для определения полных возможностей устройств клиента.

Другая информация о состоянии, определяющая сущность возможностей клиента в отношении частоты кадров растровых изображений и видео, может пересылаться в пакете состояния на хост, так что хост может сконфигурировать интерфейс так, чтобы он был таким эффективным или оптимальным, как того требует практика, или требуется в пределах любых системных ограничений.

Хост посылает пакеты заполнителя, когда нет (больше) пакетов данных, подлежащих пересылке в текущем подкадре, или когда хост не может переслать со скоростью передачи, достаточной, чтобы не отставать от скорости передачи данных, выбранной для прямой линии передачи данных. Так как каждый подкадр начинается с пакета заголовка подкадра, то окончание предыдущего подкадра содержит пакет (наиболее вероятно, что это пакет заполнителя), который точно заполняет предыдущий подкадр. В случае

недостатка пространства для по существу несущих данные пакетов пакет заполнителя, по всей видимости, будет последним пакетом в подкадре или в конце ближайшего предшествующего подкадра и перед пакетом заголовка подкадра. Задачей операций управления в устройстве хоста является обеспечение того, чтобы оставалось достаточное

5 пространство, остающееся в подкадре для каждого пакета, подлежащего передаче в пределах этого подкадра. Одновременно, если устройство хоста инициирует посылку пакета данных, то хост должен быть способен успешно завершить пакет такого размера в пределах кадра, не вызывая состояние нехватки данных.

В одном аспекте вариантов осуществления передача подкадров имеет два режима.

10 Одним режимом является периодический режим подкадров, или периодические временные эпохи, используемый для передачи потоков реального видео и аудио. В этом режиме длина Подкадра определяется как не равная нулю. Вторым режимом является асинхронный, или непериодический режим, при котором кадры используются для предоставления растровых данных клиенту, когда доступна новая информация. Этот режим определяется

15 посредством установки длины подкадра, равной нулю, в Пакете заголовка подкадра. При использовании периодического режима прием пакетов подкадра может начинаться тогда, когда клиент синхронизировался со структурой кадра прямой линии передачи данных. Это соответствует состояниям «в синхронизме», определяемым в соответствии с диаграммой состояния, описываемой ниже в отношении фиг.49 или фиг.63. В асинхронном

20 непериодическом режиме подкадров прием начинается после приема первого пакета Заголовка подкадра.

В. Общая структура пакета

Ниже представлены формат или структура пакетов, используемых для формулировки протокола обмена данными или передачи сигналов, или способа или средства для

25 пересылки данных, реализуемых вариантами осуществления, помня, что интерфейс является расширяемым и могут быть добавлены дополнительные структуры пакетов, когда это потребуется. Пакеты обозначаются как, или делятся на различные «типы пакетов» на основе их назначения в интерфейсе, т.е. команды, информация, значение или данные, которые они переносят или с которыми они связаны. Поэтому каждый тип пакета

30 обозначает предварительно определенную структуру пакета для данного пакета, который используется при манипулировании передаваемыми пакетами и данными. Как очевидно, пакеты могут иметь предварительно выбранную длину или иметь переменные или динамически изменяемые длины в зависимости от их соответствующих назначений. Пакеты также могут иметь различные названия, хотя реализуется, тем не менее, одинаковое

35 назначение, что может иметь место, когда протоколы изменяются во время принятия в стандарт. Байты или значения байтов, используемые в различных пакетах, конфигурируются как многобитовые (8- или 16-битовые) целые числа без знака. Краткое описание используемых пакетов вместе с их обозначениями «типов», перечисленными в порядке типов, приведено в таблице VI-1 - VI-4.

40 Каждая таблица представляет общий «тип» пакета в общей структуре пакетов для легкой иллюстрации и понимания. Нет ограничения или другого влияния, подразумеваемого или выражаемого для изобретения этими группированиями, и пакеты могут быть организованы многими другими способами, как потребуется. Также отмечается направление, в котором пересылка пакета рассматривается действительной.

45

Таблица VI-1 Пакеты управления линией передачи данных			
Название пакета	Тип пакета	Действителен в прямом направлении	Действителен в обратном направлении
Пакет заголовка подкадра	15359	x	
Пакет заполнителя	0	x	x
Пакет инкапсуляция обратной линии передачи данных	65	x	
Пакет закрытия линии передачи данных	69	x	
Пакет состояния питания дисплея	75	x	
Пакет включения прямых аудиоканалов	78	x	
Пакет выполнение перехода на тип	77	x	

50

Пакет измерения задержки на двойное прохождение	82	x	
Пакет калибровки перекоса прямой линии передачи данных	83	x	

5

Таблица VI-2 Пакеты базового медиапотока			
Название пакета	Тип пакета	Действителен в прямом направлении	Действителен в обратном направлении
Пакет видеопотока	16	x	x
Пакет аудиопотока	32	x	x
Пакеты зарезервированных потоков	1-15, 18-31, 33-55	x	x
Пакеты определяемого пользователем потока	56-63	x	x
Пакет карты цветов	64	x	x
Пакет частоты аудиоотсчетов обратной линии передачи данных	79	x	
Пакет разрешения прозрачного цвета	81	x	

10

15

Таблица VI-3 Пакеты состояния и управления клиента			
Название пакета	Тип пакета	Действителен в прямом направлении	Действителен в обратном направлении
Пакет возможностей клиента	66		x
Пакет данных клавиатуры	67	x	x
Пакет данных указательного устройства	68	x	x
Пакет запроса клиента и состояния	70		x
Пакет служебных данных защиты цифрового содержимого	80	x	x
Пакет запроса свойств ВПУ	128	x	
Пакет ответа свойств ВПУ	129		x
Пакет установки свойств ВПУ	130	x	
Пакет запроса действительных параметров	131	x	
Пакет ответа действительных параметров	132		x
Пакет запроса заданного состояния	138	x	
Пакет списка ответа действительных состояний	139		x
Пакет параметров задержки на обработку пакета	140		x
Пакет возможностей персонального дисплея	141		x
Пакет отчета об ошибках клиента	142		x
Пакет возможностей масштабируемого видеопотока	143		x
Пакет идентификации клиента	144		x
Пакет возможностей альтернативного дисплея	145		x
Пакет доступа к регистрам	146	x	x

20

25

30

35

40

45

Таблица VI-4 Пакеты усовершенствованной графики и дисплея			
Название пакета	Тип пакета	Действителен в прямом направлении	Действителен в обратном направлении
Пакет пересылки битовых блоков	71	x	
Пакет заливки растровой области	72	x	
Пакет заливки растровым узором	73	x	
Пакет считывания буфера кадров	74	x	
Пакет возможностей масштабируемого видеопотока	143		x
Пакет установки масштабируемого видеопотока	136	x	
Пакет подтверждения масштабируемого видеопотока	137		x
Пакет масштабируемого видеопотока	18	x	

50

Кое-что, что ясно из других обсуждений в данном тексте, представляет собой то, что каждый Пакет заголовка подкадра, заполнителя, инкапсуляции обратной линии передачи данных, закрытия линии передачи данных, возможностей клиента и запроса клиента и состояния считается очень важным или даже требуется во многих вариантах осуществления интерфейсов обмена данными для работы во Внешнем режиме. Однако Пакеты инкапсуляции обратной линии передачи данных, закрытия линии передачи данных, возможностей клиента и запроса клиента и состояния могут быть или, более вероятно,

что они считаются необязательными для работы во Внутреннем режиме. Это создает еще другой тип протокола ЦИМД, который дает возможность выполнять обмен данными с очень высокими скоростями с уменьшенным набором пакетов обмена данными и соответствующим упрощением управления и согласованием во времени.

5 Пакеты имеют общую базовую структуру или полный набор минимальных полей, содержащий поле Длины пакета, поле Типа пакета, поле (поля) Байтов данных и поле ЦИК, который показан на фиг.8. Как показано на фиг.8, поле Длины пакета содержит информацию в виде многобитового или многобайтового значения, которое задает общее количество битов в пакете, или его длину между полем длины пакета и полем ЦИК. В
10 одном варианте осуществления поле длины пакета содержит 16-битовое или 2-байтовое целое число без знака, которое задает длину пакета. Поле Типа пакета представляет собой другое многобитовое поле, которое обозначает тип информации, которая содержится в пакете. В примерном варианте осуществления им является 16-битовое или 2-байтовое значение, в виде 16-битового целого числа без знака, и оно задает такие типы данных
15 как возможности дисплея, переход, видео- или аудиопотоки, состояние и т.д.

 Третьим полем является поле Байтов данных, которое содержит биты или данные, передаваемые или пересылаемые между устройствами хоста и клиента, как часть этого пакета. Формат данных определяется специально для каждого типа пакета в соответствии с заданным типом пересылаемых данных, и поле может быть разделено на группу
20 дополнительных полей, каждое со своими собственными требованиями к формату. Т.е. каждый тип пакета будет иметь определенный формат для этой части или поля. Последним полем является поле ЦИК, которое содержит результаты проверки 16-битовым циклическим избыточным кодом, вычисленные по полям Байтов данных, Типа пакета и Длины пакета, которое используется для подтверждения целостности информации в
25 пакете. Другими словами, вычисленное по всему пакету за исключением самого поля ЦИК. Клиент, как правило, сохраняет суммарное число обнаруженных ошибок ЦИК и сообщает это число обратно хосту в пакете Запроса клиента и состояния (см. описание ниже).

 Как правило, эти длины и организация полей предназначены для того, чтобы сохранять 2-байтовые поля выровненными по границе четных байтов, и 4-байтовые поля
30 выровненными по 4-байтовым границам. Это позволяет легко встраивать структуры пакетов в пространство основной памяти хоста и клиента или связывать с ними без нарушения правил выравнивания типов данных, встречаемых для большинства или обычно используемых процессоров или схем управления.

 Во время пересылки пакетов поля передаются, начиная сначала с самого младшего
35 бита (СМБ) и заканчивая самым старшим битом (ССБ), передаваемым последним. Параметры, длина которых превышает один байт, передаются с использованием сначала самого младшего байта, что приводит к той же форме передачи битов, используемой для параметра, длина которого превышает 8 битов, что и та, которая используется для более короткого параметра, где сначала передается СМБ. Поля данных каждого пакета, как
40 правило, передаются по порядку, в котором они определены в последующих разделах ниже, причем первое перечисленное поле передается первым, и последнее описанное поле передается последним. Данные на тракте сигнала MDDI_Data0 выравниваются с битом «0» байтов, передаваемых по интерфейсу в любом из режимов, Типа 1, Типа 2, Типа 3 или Типа-4.

45 При манипулировании данными для дисплеев данные массивов пикселей передаются сначала строками, затем столбцами, как традиционно выполняется в электронной технике. Другими словами, все пиксели, которые находятся в одной строке растрового изображения, передаются по порядку, при этом первым передается крайний левый пиксель и последним передается крайний правый пиксель. После того как будет передан крайний
50 правый пиксель строки, то тогда следующим пикселем в последовательности является крайний левый пиксель следующей строки. Для большинства дисплеев строки пикселей, как правило, передаются по порядку сверху вниз, хотя по необходимости могут быть использованы другие конфигурации. Кроме того, при обработке растровых изображений

обычным подходом, которому придерживаются и в данном случае, является определение опорной точки обозначением левого верхнего угла растрового изображения, как расположение или смещение «0,0». Координаты X и Y, используемые для задания или определения положения на растровом изображении, увеличиваются по значению при

5 приближении к правой и нижней части растрового изображения соответственно. Первая строка и первый столбец (верхний левый угол изображения) начинаются со значения индекса, равного нулю. Величина координаты X увеличивается к правой стороне изображения, и величина координаты Y увеличивается к нижней части изображения, как видит пользователь дисплея.

10 Окно дисплея представляет собой видимую часть растрового изображения, часть пикселей в растровом изображении, которую может видеть пользователь на физической среде дисплея. Часто бывает так, что окно дисплея и растровое изображение имеют одинаковый размер. Верхний левый угол окна дисплея всегда отображает расположение «0,0» пикселя растрового изображения. Ширина окна дисплея соответствует оси X

15 растрового изображения, и ширина окна дисплея для данного варианта осуществления меньше или равна ширине соответствующего растрового изображения. Высота окна соответствует оси Y растрового изображения, и высота окна дисплея для данного варианта осуществления меньше или равна высоте соответствующего растрового изображения. Само окно дисплея не является адресуемым в протоколе, так как оно

20 определяется только как видимая часть растрового изображения.

Зависимость между растровыми изображениями и окнами дисплея хорошо известна в компьютерной, электронной технике, обмене данных при помощи интернета и в других относящихся к электронике областях техники. Поэтому здесь не приводится дополнительное обсуждение или иллюстрация этих принципов.

25 С. Определения пакетов

1. Пакет заголовка подкадра

Пакет заголовка подкадра является первым пакетом каждого подкадра и имеет базовую структуру, изображенную на фиг.9. Пакет заголовка подкадра используется для синхронизации хоста и клиента, каждый хост должен иметь возможность генерировать

30 данный пакет, в то время как каждый клиент должен иметь возможность принимать и интерпретировать этот пакет. Как можно видеть в одном варианте осуществления по фиг.9, этот тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, Служебного слова, Резервированное 1, Длины подкадра, Версии протокола, Числа подкадров и Числа медиакадров, как правило, в этом порядке. В одном варианте

35 осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 15359 (0x3bff шестнадцатеричный) и использует предварительно выбранную фиксированную длину, равную 20 байтам, не включая поле длины пакета.

Каждый из поля Типа пакета и поля Служебного слова использует 2-байтовое значение (16-битовое целое число без знака). 4-байтовая комбинация этих двух полей вместе

40 образует 32-битовое служебное слово с хорошей автокорреляцией. В одном варианте осуществления фактическим служебным словом является 0x005a3bff, где сначала передаются младшие 16 битов в качестве Типа пакета, а после этого передаются самые старшие 16 битов.

Поле Резервированное 1 содержит 2 байта, которые резервируют пространство для

45 будущего использования и, как правило, конфигурируются в данный момент со всеми битами, установленными в нуль. Одним назначением этого поля является то, чтобы вызывать выравнивание последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов. Самый младший бит резервируется для указания, имеет ли или нет хост возможность адресации

50 многочисленных устройств клиента. Значение нуль для этого байта резервируется для указания того, что хост имеет возможность работы только с одним устройством клиента.

Поле Длины подкадра содержит 4 байта информации, или значений, которые задают количество байтов на подкадр. В одном варианте осуществления длина данного поля

может быть установлена равной нулю для указания того, что только один подкадр будет передаваться хостом перед закрытием линии передачи данных и переходом ее в состояние незанятости. Значение в этом поле может быть изменено динамически «на лету» во время перехода от одного подкадра к следующему. Эта возможность полезна для того, чтобы

5 выполнять незначительные подстройки временных соотношений синхрои импульсов для обеспечения изохронных потоков данных. Если не является действительным ЦИК пакета Заголовка подкадра, то тогда контроллер линии передачи данных должен использовать Длину подкадра предыдущего заведомо хорошего пакета Заголовка подкадра для оценки длины текущего подкадра.

10 Поле Версии протокола содержит 2 байта, которые задают версию протокола, используемую хостом. Поле Версии протокола устанавливается в «0» для задания первой или текущей версии используемого протокола. Это значение будет изменяться во времени по мере создания новых версий и уже модифицируется до значения «1» для некоторых полей версии. Значения версий вероятно или в основном будут придерживаться номера

15 текущей версии для документа одобренных стандартов, который охватывает интерфейсы, такие как ЦИМД, как будет известно.

Поле Числа подкадров содержит 2 байта, которые задают порядковый номер, указывающий количество подкадров, которые были переданы с начала медиакадра. У первого подкадра медиакадра Число подкадров равно нулю. Значение последнего

20 подкадра медиакадра равно $n-1$, где n - количество подкадров в медиакадре. Значение поля Числа подкадров равно Числу подкадров, посланному в предыдущем пакете Подкадра плюс 1, за исключением первого подкадра медиакадра, который будет иметь число нуль. Заметьте, что, если Длина подкадра установлена в нуль (указывая на непериодический подкадр), то тогда число Подкадров также устанавливается равным нулю.

25 Поле Числа медиакадров содержит 4 байта (32-битовое целое число без знака), которые задают порядковый номер, указывающий количество медиакадров, которые были переданы с начала текущего передаваемого медиаэлемента или медиаданных. Первый медиакадр медиаэлемента имеет Число медиакадров, равное нулю. Число медиакадров получает приращение перед первым подкадром каждого медиакадра и возвращается обратно к нулю

30 после использования максимального Числа медиакадров (например, число медиакадров $2^{31}-1=4294967295$). Значение Числа медиакадров может быть сброшено, как правило, в любое время Хостом для выполнения требований конечного применения.

2. Пакет Заполнителя

Пакетом заполнителя является пакет, который пересылается на устройство клиента, или

35 с него, когда нет другой доступной информации, подлежащей пересылке или по прямой, или по обратной линии передачи данных. Рекомендуются, чтобы пакеты заполнителя имели минимальную длину для достижения максимальной гибкости при посылке других пакетов, когда это требуется. В самом конце подкадра или пакета инкапсуляции обратной линии передачи данных (см. ниже) контроллер линии передачи данных устанавливает размер

40 пакета заполнителя таким, чтобы он заполнял остающееся пространство для поддержания целостности пакета. Пакет заполнителя является полезным для сохранения согласования во времени на линии передачи данных, когда хост или клиент не имеют информации для посылки или обмена. Каждому хосту и клиенту необходимо посылать и принимать данный пакет, чтобы сделать эффективным использование интерфейса.

45 Примерный вариант осуществления формата и содержимого Пакета заполнителя показаны на фиг.10. Как показано на фиг.10, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, Байтов заполнителя и ЦИК. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как Тип 0, что указывается в 2-байтовом поле Типа. Биты или байты в поле Байтов заполнителя содержат

50 переменное количество значений битов, всех равных нулю, чтобы получить требуемую длину пакета заполнителя. Наименьший пакет заполнителя не содержит байтов в данном поле. Т.е. пакет состоит только из полей длины пакета, типа пакета и ЦИК и в одном варианте осуществления использует предварительно выбранную фиксированную длину,

равную 6 байтам, или значение Длины пакета, равное 4. Значение ЦИК определяется для всех байтов в пакете, включая поле Длины пакета, которое может исключаться в некоторых других типах пакетов.

3. Пакет видеопотока

- 5 Пакеты Видеопотока содержат видеоданные для обновления обычно прямоугольных областей устройства дисплея. Размер этой области может быть минимально равен одному пикселю или максимально - всему дисплею. Может быть почти неограниченное количество одновременно отображаемых потоков, ограничиваясь системными ресурсами, потому что весь контекст, необходимый для отображения потока, содержится в Пакете Видеопотока.
- 10 Формат одного варианта осуществления Пакета Видеопотока (Дескриптор формата видеоданных) показан на фиг.11. Как показано на фиг.11, в одном варианте осуществления данный тип пакета структурирован так, что имеет поля Длины пакета (2 байта), Типа пакета, ИД bClient, Дескриптора видеоданных, Атрибутов пикселей дисплея, Левого края по оси X, Верхнего края по оси Y, Правого края по оси X, Нижнего края по
- 15 оси Y, Начала X и Y, Числа пикселей, ЦИК параметров, Данных пикселей и ЦИК данных пикселей. Данный тип пакета, как правило, идентифицируется как Тип 16, который указывается в 2-байтовом поле Типа. В одном варианте осуществления клиент указывает возможность приема Пакета видеопотока, используя поля Возможностей формата RGB (красный, зеленый, синий), монохромного формата и формата Y Cr Cb Пакета
- 20 возможностей клиента.

- В одном варианте осуществления поле ИД bClient содержит 2 байта информации, которые резервируются для ИД клиента. Так как это вновь разработанный протокол обмена данными, фактические ИД клиента еще не являются известными или не являются достаточно обмениваемыми. Поэтому биты в этом поле, как правило, устанавливаются
- 25 равными нулю до тех пор, пока такие значения ИД не будут известны, в то время значения ИД могут быть вставлены или использованы, что является очевидным для специалиста в данной области техники. Этот же процесс в основном будет действительным для полей ИД клиента, описанных ниже.

- На фиг.12A-12E показаны формат и содержимое, используемые для реализации
- 30 принципа действия примерного поля Дескриптора видеоданных, как упомянуто выше. На фиг.12A-12E поле Дескриптора формата видеоданных содержит 2 байта в виде 16-битового целого числа без знака, которое задает формат каждого пикселя в Данных пикселей в текущем потоке текущего пакета. Возможно, что различные пакеты Видеопотока могут использовать различные форматы данных пикселей, т.е. использовать различные значения
- 35 в поле Дескриптора формата видеоданных, и аналогично поток (область дисплея) может изменять свой формат данных «на лету». Формат данных пикселей должен соответствовать по меньшей мере одному из действительных форматов для клиента, определенных в Пакете возможностей клиента. Дескриптор формата видеоданных определяет формат пикселя только для текущего пакета, что не означает, что постоянный
- 40 формат будет продолжаться использоваться во время существования конкретного видеопотока.

- На фиг.12A-12D изображено то, как кодируется Дескриптор формата видеоданных. Как использовано на этих фигурах и в данном варианте осуществления, когда биты [15:13] равны «000», как показано на фиг.12A, тогда видеоданные состоят из массива пикселей
- 45 монохромного формата, где количество битов на пиксель определяется битами 3-0 слова Дескриптора формата видеоданных. Биты 11-4, как правило, резервируются для будущего использования или применений и устанавливаются в нуль в данной ситуации. Когда биты [15:13] вместо этого равны значениям «001», как показано на фиг.12B, тогда видеоданные состоят из массива цветных пикселей, каждый из которых задает цвет
- 50 посредством карты цветов (палитры). В данном случае биты 5-0 слова Дескриптора формата видеоданных определяют количество битов на пиксель, и биты 11-6, как правило, резервируются для будущего использования или применений и устанавливаются равными нулю. Когда вместо этого биты [15:13] равны значениям «010», как показано на фиг.12C,

тогда видеоданные состоят из массива цветных пикселей, где количество битов на пиксель красного цвета определяется битами 11-8, количество битов на пиксель зеленого цвета определяется битами 7-4 и количество битов на пиксель синего цвета определяется битами 3-0. В данном случае суммарное количество битов в каждом пикселе представляет собой сумму количества битов, используемых для красного, зеленого и синего цветов.

Однако когда вместо этого биты [15:13] равны значениям или строке «011», как показано на фиг.12D, тогда видеоданные состоят из массива видеоданных в формате 4:2:2 YCbCr с яркостной и цветовой информацией, где количество битов на пиксель яркости (Y) определяется битами 11-8, количество битов составляющей Cb определяется битами 7-4 и количество битов составляющей Cr определяется битами 3-0. Общее количество битов в каждом пикселе представляет собой сумму количества битов, используемых для красного, зеленого и синего цветов. Составляющие Cb и Cr пересылаются с половинной скоростью относительно Y. Кроме того, видеотсчеты в части Данных пикселей этого пакета организованы следующим образом: Cbn, Yn, Crn, Yn+1, Cbn+2, Yn+2, Crn+2, Yn+3..., где Cbn и Crn связаны с Yn и Yn+1, и Cbn+2 и Crn+2 связаны с Yn+2 и Yn+3 и т.д.

Yn, Yn+1, Yn+2 и Yn+3 представляют собой значения яркости для четырех последовательных пикселей в одной строке слева направо. Если существует нечетное количество пикселей в строке (Правый край по оси X - Левый край по оси X+1) в окне, указываемым Пакетом видеопотока, тогда за значением Y, соответствующим последнему пикселю в каждой строке, будет следовать значение Cb первого пикселя следующей строки и значение Cr не посылается для последнего пикселя в строке. Рекомендуется, чтобы окно, использующее формат Y Cb Cr, имело ширину, которая составляет четное число пикселей. Данные пикселей в пакете должны содержать четное количество пикселей. Они могут содержать нечетное или четное количество пикселей в том случае, когда последний пиксель Данных пикселей соответствует последнему пикселю строки в окне, заданным в заголовке Пакета видеопотока, т.е. когда расположение X последнего пикселя в Данных пикселей равно Правому краю по оси X.

Когда биты [15:13] вместо этого равны значениям «100», тогда видеоданные состоят из массива пикселей байеровского формата, где количество битов на пиксель определяется битами 3-0 слова Дескриптора формата видеоданных. Узор группы пикселей определяется битами 5 и 4, как показано на фиг.12E. Порядок данных пикселей может быть горизонтальным или вертикальным, и пиксели в строках или столбцах могут посылаться в прямом или обратном порядке, и определяется битами 8-6. Биты 11-9 должны быть установлены в нуль. Группа из четырех пикселей в группе пикселей в байеровском формате имеет сходство с тем, что часто упоминается как единственный пиксель в некоторых технологиях изготовления дисплеев. Однако один пиксель в байеровском формате представляет собой только один из четырех цветных пикселей из узора мозаики группы пикселей.

Для всех пяти форматов, показанных на фигурах, Бит 12, который обозначается как «Р», задает, являются ли или нет отсчеты Данных пикселей упакованными или выровненными по байтам данными пикселей. Значение «0» в этом поле указывает, что каждый пиксель в поле Данных пикселей выровнен по байтам с границей байтов ЦИМД. Значение «1» указывает, что каждый пиксель и каждый цвет в каждом пикселе в Данных пикселей является упакованным по отношению к предыдущему пикселю или цвету в пикселе, не оставляя неиспользованных битов. Разность между форматом Выровненных по байтам и Упакованных данных пикселей подробно показана на фиг.13, где можно ясно видеть, что выровненные по байтам данные могут оставлять неиспользуемые части подкадра данных в противоположность упакованному формату пикселей, который не оставляет.

4. Пакет аудиопотока

Пакеты аудиопотока переносят аудиоданные, подлежащие воспроизведению аудиосистемой клиента, или для отдельного устройства представления аудио. Различные потоки аудиоданных могут быть назначены отдельным аудиоканалам в звуковой системе,

например: левый-передний, правый-передний, центральный, левый-задний и правый-задний, в зависимости от типа используемой аудиосистемы. Предусматриваются полное дополнение аудиоканалов для головных телефонов, которые содержат улучшенную обработку пространственно-акустического сигнала. Клиент указывает возможность приема

5 Пакета аудиопотока, используя поля Возможностей аудиоканала и Частоты аудиоотсчетов Пакета возможностей клиента. Формат Пакетов аудиопотока изображен на фиг.14.

Как показано на фиг.14, данный тип пакета структурирован в одном варианте осуществления так, что имеет поля Длины пакета, Типа пакета, ИД bClient, ИД аудиоканала, Резервированное 1, Числа аудиоотсчетов, Битов на отсчет и упаковки,

10 Частоты аудиоотсчетов, ЦИК параметров, Цифровых аудиоданных и ЦИК аудиоданных. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 32.

Поле ИД bClient содержит 2 байта информации, которая резервируется для ИД клиента, как используется ранее. Поле Резервированное 1 содержит 2 байта, которые

15 резервируются для будущего использования и в основном конфигурируются в данный момент со всеми битами, установленными в нуль.

Поле Битов на отсчет и упаковки содержит 1 байт в виде 8-битового целого числа без знака, которое задает формат упаковки аудиоданных. В обычно используемом формате Биты 4-0 определяют количество битов на ИКМ-аудиоотсчет. Бит 5 тогда задает,

20 упаковываются ли или нет отсчеты Цифровых аудиоданных. Разница между упакованными и выровненными по байтам аудиоотсчетами, использующими здесь 10-битовые отсчеты, изображена на фиг.15. Значение «0» указывает, что каждый ИКМ-аудиоотсчет в поле Цифровых аудиоданных выравнивается по байтам с границей байтов ЦИМД, и значение «1» указывает, что каждый последующий ИКМ-аудиоотсчет упаковывается по отношению к

25 предыдущему аудиоотсчету. Этот бит, как правило, является эффективным только тогда, когда значение, определенное в битах 4-0 (количество битов на ИКМ-аудиоотсчет), не является кратным восьми. Биты 7-6 резервируются для будущего использования и, как правило, устанавливаются в значение нуль.

5. Пакеты зарезервированных потоков

В одном варианте осуществления типы 1-15, 18-31 и 33-55 пакетов резервируются для пакетов потоков, подлежащих определению для использования в будущих версиях или модификациях пакетных протоколов, когда потребуется для различных встречаемых применений. Также это является частью выполнения ЦИМД более гибким и полезным в

30 отношении ко все изменяющейся технологии и разработкам системы по сравнению с другими методами.

6. Пакеты Определяемого пользователем потока

Восемь типов потоков данных, известных как Типы 56-63, резервируются для использования в оригинальных применениях, которые могут быть определены изготовителями оборудования для использования с линией передачи данных ЦИМД. Они

40 известны как Пакеты Определяемого пользователем потока. Такие пакеты могут использоваться для любой цели, но хост и клиент должны использовать такие пакеты только в тех случаях, когда результат такого использования является очень хорошо понимаемым или известным. Конкретное определение параметров и данных потоков для этих типов пакетов оставляется изготовителям конкретного оборудования и разработчикам интерфейсов, реализующих такие типы пакетов или добывающихся их применения.

45 Некоторые примерные использования Пакетов определяемого пользователем потока предназначены для передачи тестовых параметров и тестовых результатов, данных заводской калибровки и оригинальных данных специального использования. Формат пакетов определяемого пользователем потока, используемый в одном варианте

50 осуществления, изображен на фиг.16. Как показано на фиг.16, данный тип пакета структурирован так, что имеет поля Длины пакета (2 байта), Типа пакета, номера ИД bClient, Параметров потока, ЦИК параметров, Данных потока и ЦИК данных потока.

7. Пакеты карты цветов

Пакеты карты цветов задают содержимое таблицы преобразования карты цветов, используемой для представления цветов для клиента. Некоторые применения могут потребовать карту цветов, которая больше, чем количество данных, которое может быть передано в одном пакете. В этих случаях могут пересылаться многочисленные пакеты

5 Карты цветов, при этом каждый с различным поднабором карты цветов посредством использования полей смещения и длины, описанных ниже. Формат Пакета карты цветов в одном варианте осуществления изображен на фиг.17. Как показано на фиг.17, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Числа элементов карты цветов, Смещения карты цветов, ЦИК параметров, Данных карты
10 цветов и ЦИК данных. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 64 (Пакет формата видеоданных и карты цветов), как задано в Поле типа пакета (2 байта). Клиент указывает возможность приема Пакетов карты цветов, используя поля Размера карты цветов и Разрядности карты цветов Пакета возможностей клиента.

15 **8. Пакеты инкапсуляции обратной линии передачи данных**

В примерном варианте осуществления в обратном направлении данные пересылаются с использованием Пакета инкапсуляции обратной линии передачи данных. Посылается пакет по прямой линии передачи данных и работа (направление пересылки) линии передачи
20 данных ЦИМД меняется или реверсируется в середине этого пакета, так что пакеты могут пересылаться в обратном направлении. Формат пакета Инкапсуляции обратной линии передачи данных в одном варианте осуществления изображен на фиг.18. Как показано на фиг.18, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Флагов обратной линии передачи данных, Делителя обратной скорости передачи, Длины реверсирования передачи 1, Длины реверсирования передачи 2, ЦИК
25 параметров, Всех нулей 1, Реверсирования передачи 1, Пакетов обратных данных, Реверсирования передачи 2 и Всех нулей 2. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 65. Для Внешнего режима каждый хост должен иметь возможность генерировать этот пакет и принимать данные, и каждый клиент должен иметь возможность принимать и посылать данные на хост, чтобы
30 эффективно использовать требуемый протокол и результирующую скорость. Реализация данного пакета является необязательной для Внутреннего режима, но Пакет инкапсуляции обратной линии передачи данных используется для того, чтобы хост принимал данные от клиента.

Контроллер линии передачи данных ЦИМД работает особым образом при посылке
35 Пакета инкапсуляции обратной линии передачи данных. ЦИМД имеет строб-сигнал, который в основном всегда возбуждается хостом в качестве контроллера линии передачи данных. Хост работает так, как если бы он передавал нуль для каждого бита частей Реверсирования передачи и Пакетов обратных данных пакета Инкапсуляции обратной линии передачи данных. Хост переключает сигнал MDDI_Strobe на каждой границе бита во
40 время двух периодов времени реверсирования передачи и во время, выделенное для пакетов обратных данных. Т.е. хост переключает MDDI_Stb с начала поля Всех нулей 1 до окончания поля Всех нулей 2. (Эта работа аналогична той, как если бы он передавал данные с одними нулями.)

Хост выключает свои линейные драйверы сигнала данных ЦИМД и в основном
45 обеспечивает, чтобы они были полностью выключены до последнего бита поля Реверсирования передачи 1, и затем повторно включает свои линейные драйверы во время поля Реверсирования передачи 2 и в основном обеспечивает, чтобы драйверы были полностью повторно включены до последнего бита поля Реверсирования передачи 2. Клиент считывает параметр Длины реверсирования передачи и возбуждает сигналы
50 данных в направлении хоста непосредственно после последнего бита в поле Реверсирования передачи 1. Т.е. клиент тактирует новые данные на линию передачи данных по определенным передним фронтам строба ЦИМД, как задано в описании содержимого пакета ниже, и в другом месте. Клиент использует параметры Длины пакета и

Длины реверсирования передачи, чтобы получить сведения о продолжительности времени, предоставленном ему для посылки пакетов на хост. Клиент может послать пакеты заполнителя или возбудить линии данных нулевым состоянием, когда у него нет данных для пересылки на хост. Если линии данных возбуждаются нулем, то хост интерпретирует
 5 это как пакет с нулевой длиной (недействительная длина) и хост больше не принимает никакие другие пакеты от клиента в течение текущего пакета Инкапсуляции обратной линии передачи данных.

В одном варианте осуществления поле Запроса обратной линии передачи данных Пакета запроса клиента и состояния может использоваться для информирования хоста о
 10 количестве байтов, которое необходимо клиенту в Пакете инкапсуляции обратной линии передачи данных для посылки данных обратно на хост. Хост делает попытку удовлетворить запрос посредством выделения по меньшей мере этого количества байтов в Пакете инкапсуляции обратной линии передачи данных. Хост может послать более одного Пакета инкапсуляции обратной линии передачи данных в подкадре. Клиент может послать
 15 Пакет запроса клиента и состояния почти в любой момент времени, и хост будет интерпретировать параметр Запроса обратной линии передачи данных как суммарное количество байтов, запрашиваемых в одном подкадре.

9. Пакеты возможностей клиента

Хосту необходимо знать возможности клиента (дисплея), с которым он выполняет обмен
 20 данными, чтобы сконфигурировать линию передачи данных хост-клиент, как правило, оптимальным или требуемым образом. Рекомендуется, чтобы дисплей посылал Пакет возможностей клиента на хост после достижения синхронизации прямой линии передачи данных. Передача такого пакета считается необходимой, когда это запрашивается хостом, используя Флаги обратной линии передачи данных в Пакете инкапсуляции обратной линии
 25 передачи данных. Пакет возможностей клиента используется для информирования хоста о возможностях клиента. Для Внешнего режима каждый хост должен иметь возможность принимать этот пакет, и каждый клиент должен иметь возможность посылать этот пакет, чтобы полностью использовать данный интерфейс и протокол. Реализация данного пакета является необязательной для Внутреннего режима, так как возможности клиента, такого
 30 как дисплей, клавиатура или другое устройства ввода-вывода, в данном случае уже должно быть правильно определено и известно хосту в момент производства или сборки в единственный компонент или блок некоторого типа.

Формат пакета Возможностей клиента в одном варианте осуществления изображен на фиг.19. Как показано на фиг.19, для этого варианта осуществления данный тип пакета
 35 структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Версии протокола, Минимальной версии протокола, Возможностей скорости передачи данных, Возможностей типа интерфейса, Количества альтернативных дисплеев, Резервированное 1, Ширины растрового изображения, Высоты растрового изображения, Ширины окна дисплея, Высоты окна дисплея, Размера карты цветов, Разрядности карты цветов формата
 40 RGB, Возможностей формата RGB, Возможностей монохромного формата, Резервированное 2, Возможностей формата Y Cr Cb, Возможностей байеровского формата, Резервированное 3, Возможностей свойств клиента, Максимальной частоты видеокадров, Минимальной частоты видеокадров, Минимальной частоты подкадров, Емкости буфера аудио, Возможностей аудиоканала, Возможностей частоты аудиоотсчетов,
 45 Разрешения аудиоотсчетов, Разрешения отсчетов микрофона, Возможностей частоты отсчетов микрофона, Формата данных клавиатуры, Формата данных указательного устройства, Типа защиты содержимого, Названия изготовителя, Кода продукта, Резервированное 4, Серийного номера, Недели изготовления, Года изготовления и ЦИК. В примерном варианте осуществления данный тип пакета, как правило, идентифицируется
 50 как пакет Типа 66.

10. Пакеты данных клавиатуры

Пакет данных клавиатуры используется для посылки данных клавиатуры с устройства клиента на хост. Может использоваться беспроводная (или проводная) клавиатура

совместно с различными дисплеями или аудиоустройствами, включая в себя, но не ограничиваясь ими, видеодисплей/устройство представления аудио, установленное на голове. Пакет данных клавиатуры передает данные клавиатуры, принимаемые от одного из нескольких известных клавиатуроподобных устройств, на хост. Этот пакет также может использоваться на прямой линии передачи данных для посылки данных на клавиатуру. Клиент указывает возможность посылки и приема Пакетов данных клавиатуры, используя Поле данных клавиатуры в Пакете возможностей клиента.

Формат Пакета данных клавиатуры показан на фиг.20 и содержит изменяемое количество байтов информации с клавиатуры или для нее. Как показано на фиг.20, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД bClient, Формата данных клавиатуры, Данных клавиатуры и ЦИК. Здесь данный тип пакета, как правило, идентифицируется как пакет Типа 67.

ИД bClient представляет собой зарезервированное поле, как и раньше, и ЦИК выполняется по всем байтам пакета. Поле Формата данных клавиатуры содержит 2-байтовое значение, которое описывает формат данных клавиатуры. Биты 6-0 должны быть идентичными полю Формата данных клавиатуры в Пакете возможностей клиента. Это значение не должно равняться 127. Биты 15-7 резервируются для будущего использования и, поэтому в настоящее время устанавливаются в нуль.

11. Пакеты данных указательного устройства

Пакет данных указательного устройства используется в качестве способа, структуры или средства для посылки информации о положении с беспроводной мыши или другого указательного устройства с клиента на хост. Данные также могут посылаться на указательное устройство по прямой линии передачи данных, используя этот пакет. Примерный формат Пакета данных указательного устройства показан на фиг.21 и содержит изменяемое количество байтов информации с указательного устройства или для него. Как показано на фиг.21, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД bClient, Формата указательного устройства, Данных указательного устройства и ЦИК. В примерном варианте осуществления этот тип пакета, как правило, идентифицируется как пакет Типа 68 в 1-байтовом поле типа.

12. Пакеты закрытия линии передачи данных

Пакет закрытия линии передачи данных посылается с хоста клиенту в качестве способа или средства для указания того, что данные и строб ЦИМД будут выключены и переведены в состояние «спячки» с малой потребляемой мощностью. Этот пакет полезен для закрытия линии передачи данных и экономии потребляемой мощности после пересылки неподвижных растровых изображений с мобильного устройства обмена данными на дисплей, или когда в настоящий момент больше нет информации для пересылки с хоста на клиент. Нормальная работа возобновляется, когда хост снова посылает пакеты. Первым пакетом, посылаемым после «спячки», является пакет заголовка подкадра. Формат Пакета состояния клиента для одного варианта осуществления показан на фиг.22. Как показано на фиг.22, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ЦИК и Всех нулей. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 69 в 1-байтовом поле типа.

Поле длины пакета использует 2 байта для задания общего количества байтов в пакете, не включая поле длины пакета. В одном варианте осуществления Длина пакета данного пакета зависит от Типа интерфейса или режима линии передачи данных, действующего в тот момент, когда посылается Пакет закрытия линии передачи данных. Поэтому типовая длина пакета становится равной 20 байтам для режима Типа 1 (22 байта в сумме в пакете), 36 байтам для режима Типа 2 (38 байтов в сумме в пакете), 68 байтам для линии передачи данных в режиме Типа 3 (70 байтов в сумме в пакете) и 132 байтам для режима Типа 4 (с 134 байтами в сумме в пакете).

Поле Всех нулей использует переменное количество байтов для обеспечения того, что сигналы MDDI_Data находятся на уровне логического нуля в течение достаточного времени, чтобы позволить клиенту начать восстановление тактовых импульсов, используя

только MDDI_Stb перед выключением линейных драйверов хоста. Длина поля Всех нулей зависит от Типа интерфейса или режима работы линии передачи данных, действующего в тот момент, когда посылается Пакет закрытия линии передачи данных. Длина поля Всех нулей предназначена для получения 64 импульсов на MDDI_Stb для любой установки Типа интерфейса. Поэтому длина поля Всех нулей для каждого типа интерфейса становится

5

равной 16 байтам для Типа 1, 32 байтам для Типа 2, 64 байтам для Типа 3 и 128 байтам для Типа 4.

Поле ЦИК использует 2 байта, которые содержат 16-битовое ЦИК байтов с Длины пакета до Типа пакета.

10

В состоянии «спячки» с малой потребляемой мощностью драйвер MDDI_Data0 выключается в высокоимпедансное состояние, начиная после 16-48-го цикла или импульса MDDI_Stb после последнего бита поля Всех нулей. Для линий передачи данных Типа-2, Типа-3 или Типа-4 сигналы MDDI_Data1 - MDDI_DataPwr7 также переводятся в высокоимпедансное состояние в тот же момент времени, когда выключается драйвер

15

MDDI_Data0, так как хост или клиент могут вызвать «пробуждение» линии передачи данных ЦИМД из состояния «спячки», как описано в другом месте, что представляет собой ключевое улучшение и преимущество настоящего изобретения.

Как описано в определении поля Всех нулей, MDDI_Stb переключается в течение 64 циклов, следующих за ССБ поля ЦИК Пакета закрытия линии передачи данных, чтобы способствовать организованному закрытию в контроллере клиента. Один цикл представляет собой переход из низкого в высокое состояние, за которым следует переход из высокого в низкое состояние, или переход из высокого в низкое состояние, за которым следует переход из низкого в высокое состояние. После отправки поля Всех нулей выключается драйвер MDDI_Stb в хосте.

25

13. Пакеты запроса клиента и состояния

Хосту необходимо небольшое количество информации от клиента, чтобы он мог сконфигурировать линию передачи данных хост-клиент в основном оптимальным образом. Рекомендуется, чтобы клиент посылал один Пакет запроса клиента и состояния на хост в каждом подкадре. Клиент должен посылать этот пакет в качестве первого пакета в Пакете инкапсуляции обратной линии передачи данных, чтобы гарантировать его надежную доставку хосту. Доставка этого пакета также выполняется при запросе хостом, используя Флаги обратной линии передачи данных в Пакете инкапсуляции обратной линии передачи данных. Пакет запроса клиента и состояния используется для того, чтобы сообщать хосту ошибки и состояние. Для работы во внешнем режиме каждый хост должен иметь

30

возможность принимать этот пакет, и каждый клиент должен иметь возможность посылать этот пакет, чтобы правильно или оптимально использовать протокол ЦИМД. Хотя это также рекомендуется, чтобы для внутренней работы, т.е. внутренних хостов и внутренних клиентов, должна быть поддержка этого пакета, это не требуется.

35

Формат Пакета запроса клиента и состояния показан на фиг.23. Как показано на фиг.23, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД сClient, Запроса обратной линии передачи данных, Изменения возможностей, Занятости клиента, Числа ошибок ЦИК и ЦИК. Этот тип пакета, как правило, идентифицируется как пакет Типа 70 в 1-байтовом поле типа и типично использует предварительно выбранную фиксированную длину, равную 12 байтам.

40

Поле Запроса обратной линии передачи данных может использоваться для информирования хоста о количестве байтов, которое необходимо клиенту в Пакете инкапсуляции обратной линии передачи данных для отправки данных обратно на хост. Хост должен предпринять попытку удовлетворения запроса посредством выделения по меньшей мере этого количества байтов в Пакете инкапсуляции обратной линии передачи данных. Хост может послать более одного Пакета инкапсуляции обратной линии передачи данных в подкадре, чтобы разместить данные. Клиент может послать Пакет запроса клиента и состояния в любой момент времени, и хост будет интерпретировать параметр Запроса обратной линии передачи данных как суммарное количество байтов, запрашиваемых в

50

одном подкадре. Ниже показаны дополнительные подробности и конкретные примеры того, как данные обратной линии передачи данных посылаются обратно на хост.

14. Пакеты пересылки битовых блоков

Пакет пересылки битовых блоков обеспечивает средство, структуру или способ для прокрутки областей дисплея в любом направлении. Дисплеи, которые обладают такой возможностью, сообщают об этой возможности битом 0 поля Индикаторов возможностей свойств дисплея Пакета возможностей клиента. Формат для одного варианта осуществления Пакета пересылки битовых блоков показан на фиг.24. Как показано на фиг.24, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Верхнего левого значения X, Верхнего левого значения Y, Ширины окна, Высоты окна, Перемещения окна по оси X, Перемещения окна по оси Y и ЦИК. Данный тип пакета, как правило, идентифицируется как пакет Типа 71 и в одном варианте осуществления использует предварительно выбранную фиксированную длину, равную 15 байтам.

Поля используются для задания значений координат X и Y верхнего левого угла окна, подлежащего перемещению, ширины и высоты окна, подлежащего перемещению, и количества пикселей, на которое окно должно быть перемещено горизонтально и вертикально соответственно. Положительные значения последних двух полей вызывают перемещение окна вправо и вниз, а отрицательные значения вызывают перемещение влево и вверх соответственно.

15. Пакеты заливки растровой области

Пакет заливки растровой области обеспечивает средство, структуру или способ для легкой инициализации области дисплея одним цветом. Дисплеи, которые обладают такой возможностью, сообщают об этой возможности битом 1 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента. Один вариант осуществления для формата Пакета заливки растровой области показан на фиг.25. Как показано на фиг.25, в этом случае данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Верхнего левого значения X, Верхнего левого значения Y, Ширины окна, Высоты окна, Дескриптора формата данных, Значения пикселя заливки области и ЦИК. Данный тип пакета, как правило, идентифицируется как пакет Типа 72 в 1-байтовом поле типа и использует предварительно выбранную фиксированную длину, равную 17 байтам.

16. Пакеты заливки растровым узором

Пакет заливки растровым узором обеспечивает средство или структуру для легкой инициализации области дисплея предварительно выбранным узором. Дисплеи, которые обладают такой возможностью, сообщают об этой возможности битом 2 поля Возможностей свойств клиента Пакета возможностей клиента. Верхний левый угол узора заливки выравнивается с верхним левым углом окна, подлежащего заливке, если только горизонтальное или вертикальное смещение узора не равно нулю. Если окно, подлежащее заливке, более широкое или более высокое, чем узор заливки, тогда узор может повторяться несколько раз в горизонтальном или вертикальном направлении для заливки окна. Правая или нижняя часть последнего повторяемого узора по необходимости обрезается. Если окно меньше, чем узор заливки, тогда правая сторона или нижняя часть узора заливки может обрезаться для размещения в окне.

Если горизонтальное смещение узора не равно нулю, тогда пиксели между левой стороной окна и левой стороной плюс горизонтальное смещение узора заливаются крайними правыми пикселями узора. Горизонтальное смещение узора должно быть меньше ширины узора. Аналогично, если вертикальное смещение узора не равно нулю, тогда пиксели между верхней стороной окна и верхней стороной плюс вертикальное смещение узора заливаются крайними нижними пикселями узора. Вертикальное смещение узора должно быть меньше высоты узора.

Один вариант осуществления для формата Пакета заливки растровым узором показан на фиг.26. Как показано на фиг.26, данный тип пакета структурирован так, что имеет

поля Длины пакета, Типа пакета, ИД hClient, Верхнего левого значения X, Верхнего левого значения Y, Ширины окна, Высоты окна, Ширины узора, Высоты узора, Горизонтального смещения узора, Вертикального смещения узора, Дескриптора формата данных, ЦИК параметров, Данных пикселей узора и ЦИК данных пикселей. В некоторых вариантах осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 73 в 1-байтовом поле типа.

17. Пакеты канала данных линии обмена данными

Пакет канала данных линии обмена данными обеспечивает структуру, средство или способ для клиента, имеющего вычислительную возможность высокого уровня, такого как ПЦП, для выполнения обмена данными с беспроводным передатчиком, таким как сотовый телефон или беспроводное устройство порта данных. В данном случае линия передачи данных ЦИМД действует как обычный высокоскоростной интерфейс между устройством обмена данными и вычислительным устройством с мобильным дисплеем, где этот пакет транспортирует данные на канальном уровне операционной системы для устройства. Например, этот пакет мог бы использоваться, если веб-браузер, клиент электронной почты или весь ПЦП был бы встроен в мобильный дисплей. Дисплеи, которые обладают такой возможностью, сообщают об этой возможности битом 3 поля Возможностей свойств клиента Пакета возможностей клиента.

Формат варианта осуществления для Пакета канала данных линии обмена данными показан на фиг.27. Как показано на фиг.27, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, ЦИК параметров, Данными линии обмена данными и ЦИК данных обмена. В одном варианте осуществления данный тип пакета, как правило, идентифицируется как пакет Типа 74 в поле типа.

18. Пакеты состояния питания дисплея

Пакет состояния питания дисплея обеспечивает структуру, средство или способ для размещения конкретных управляемых клиентом или связанных с клиентом, подключенных аппаратных средств или аппаратных средств контроллера в состояние с малым потреблением мощности, когда клиент, такой как дисплей, не используется или находится в текущем активном использовании, чтобы минимизировать потребление мощности системой или истощение системных ресурсов. Пакет такого типа является наиболее полезным для применений интерфейса или структуры интерфейса и протокола в конфигурациях или работе во внешнем режиме. В таких применениях более вероятно, что внешнее устройство работает с ограниченными ресурсами питания, такими как батареи, или имеет другие ограничения на питание или отношение к нему, например перегрев в ограниченных пространствах и т.п., так что требуется минимальное рабочее состояние для периодов бездействия или неиспользования. В одном варианте осуществления клиент указывает возможность ответа на Пакеты состояния питания дисплея, используя бит 9 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента.

Формат одного варианта осуществления для Пакета состояния питания дисплея показан на фиг.28. Как показано на фиг.28, в одном варианте осуществления данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Состояния питания и ЦИК. Данный тип пакета, как правило, идентифицируется как пакет Типа 75 в 2-байтовом поле типа. 2-байтовое поле ИД hClient содержит информацию или значения, которые резервируются для ИД клиента, используемого, как указано ранее. Так как это поле, как правило, резервируется для будущего использования, текущее значение устанавливается на нуль посредством установки битов в «0», хотя оно может использоваться специалистом в данной области техники для пересылки требуемой информации.

Поле состояния питания, в данном случае 2 байта, задает информацию, используемую для установки заданного устройства, части аппаратных средств или оборудования, связанного с клиентом, такого как дисплей, в заданное состояние питания. При использовании для дисплеев Бит 0 этого поля задает, применяется ли или нет пакет к основному дисплею или к альтернативному дисплею. Если бит 0 равен 1, тогда пакет

применяется к основному дисплею. Если бит 0 равен 0, тогда пакет применяется к альтернативному дисплею, задаваемому битами 11-8. Бит 1 резервируется для будущего использования и, как правило, устанавливается в ноль.

Биты 3-2 поля Состояния питания задают состояние питания дисплея, выбираемого битами 11-8 и битом 0. Когда Биты [3:2] имеют значение «00», выбранный дисплей не освещается и должен потреблять минимальную величину мощности, и не гарантируется, что содержимое буфера кадров будет сохраняться во время этого состояния. Когда Биты [3:2] имеют значение «01», выбранный дисплей не освещается и потребляет минимальную величину мощности, и гарантируется, что содержимое буфера кадров будет сохраняться во время этого состояния. Дисплей может потреблять больше мощности в этом состоянии, чем в состоянии 00. Клиент может указывать возможность поддержки состояния 01 при помощи бита 10 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента. Когда Биты [3:2] поля Состояния питания имеют значение «10», выбранный дисплей освещается и отображает изображение из связанного с ним буфера кадров. Значение «11» для Битов [3:2] представляет собой зарезервированное значение или состояние для будущего использования и не используется.

Специалисту в данной области техники понятно, что, хотя наиболее полезно для дисплейных применений, использование этого пакета не ограничивается данным изобретением только дисплеями и могут быть другие применения, конфигурации или ситуации, в которых может быть необходимым или требоваться управление питанием в отношении других аппаратных элементов, с которыми используется ЦИМД, или которые клиент управляет или с которыми он выполняет обмен данными. В этих случаях описанные выше Биты могут иметь подобные функции, но могут активизировать основные и второстепенные из таких элементов, или устанавливать уровни питания и т.п., что понятно.

Биты 11-8 поля Состояния питания образуют 4-битовое целое число без знака, которое задает альтернативный дисплей, к которому применяется состояние питания. Бит 0 устанавливается на значение логического нуля, чтобы клиент интерпретировал биты 11-8 в качестве номера альтернативного дисплея. Если бит 0 равен 1, тогда биты 11-8 равны нулю.

Биты 7-4 и Биты 15-12 резервируются для будущего использования и, как правило, устанавливаются на уровень или значения логического нуля для текущих применений или разработок.

2-байтовое поле ЦИК задает или содержит ЦИК всех байтов в пакете, включая Длину пакета.

19. Пакеты выполнения перехода на тип

Пакет выполнения перехода на тип представляет собой средство, структуру или способ для использования хостом, чтобы подавать команду клиенту на переход в режим, задаваемый в данном пакете. Это представляет собой одну из установок типа интерфейса, поддерживаемых клиентом, как описано в Пакете возможностей клиента. Хост и клиент должны переключиться на задаваемый тип интерфейса прямой и обратной линии передачи данных непосредственно после того, как посылается этот пакет. Формат одного варианта осуществления для Пакета выполнения перехода на тип показан на фиг.29. Хосты и клиенты, которые поддерживают тип интерфейса кроме Типа 1, должны обеспечивать поддержку этого пакета. Обычно рекомендуется, чтобы хост считывал Пакет запроса клиента и состояния непосредственно перед тем, как он посылает Пакет выполнения перехода на тип, для подтверждения того, что клиент находится в синхронизме с хостом.

Как показано на фиг.29, в одном варианте осуществления данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, Типа интерфейса, Резервированное 1, Заполнителя задержки и ЦИК. Данный тип пакета, как правило, идентифицируется как пакет Типа 77 в 2-байтовом поле типа и использует предварительно выбранную фиксированную длину, равную 6 байтам, вне полей Длины пакета и Заполнителя задержки.

В одном варианте осуществления поле Типа интерфейса использует 1-байтовое значение для подтверждения нового типа интерфейса, подлежащего использованию или применению для линии передачи данных. Значение в этом поле задает или представляет тип интерфейса следующим образом. Биты 2-0 определяют Тип интерфейса, подлежащий
 5 использованию на прямой линии передачи данных, причем значение 1 означает или задает переход в режим Типа 1; значение 2 - переход в режим Типа 2, значение 3 - переход в режим Типа 3, и значение 4 - переход в режим Типа 4. Биты 5-3 определяют Тип интерфейса, подлежащий использованию на обратной линии передачи данных, причем значение 1 означает или задает переход в режим Типа 1, значение 2 - переход в режим
 10 Типа 2, значение 3 - переход в режим Типа 3, и значение 4 - переход в режим Типа 4. Биты 0, 6 и 7 в настоящее время резервируются для будущего использования и как таковые обычно но необязательно устанавливаются на уровень логического нуля.

Поле Заполнителя задержки было создано как средство, структура или способ для предоставления возможности подготовки или конфигурирования достаточного времени со
 15 стороны системы для клиента для переключения на использование или установления на использование новых установок типа интерфейса в начале пакета, который следует непосредственно после Пакета выполнения перехода на тип интерфейса. Это поле содержит группу байтов или 8-битовых значений, которые все устанавливаются или равны уровню или значению логического нуля. Количество байтов, используемых в данном поле,
 20 выбирается таким, что оно приводит к тому, что длина этого поля эквивалентна 64 циклам MDDI_Stb. Длина поля Заполнителя задержки основывается на установке типа интерфейса прямой линии передачи данных, которая равна 16 байтам для типа интерфейса прямой линии передачи данных Типа 1, 32 байтам для типа интерфейса Типа 2, 64 байтам для типа интерфейса Типа 3, и 128 байтам при задании или использовании
 25 типа интерфейса прямой линии передачи данных Типа 4.

Поле Резервированное 1 (в данном случае 1 байт) резервируется для будущего использования при сообщении информации. Все биты в этом поле, как правило, устанавливаются на уровень логического нуля. Назначением таких полей в настоящее время является то, чтобы вызывать выравнивание всех последующих 2-байтовых полей с
 30 адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов. Поле ЦИК (в данном случае 2 байта) содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

20. Пакеты включения прямых аудиоканалов

Этот пакет обеспечивает структуру, способ или средство, которые позволяют хосту
 35 включать или выключать аудиоканалы в клиенте. Эта возможность является полезной, так что клиент (например, дисплей) может отключить питание от усилителей звуковой частоты или подобных схемных элементов для экономии питания, когда нет аудио для вывода хостом. Это значительно труднее осуществить неявным образом, просто используя присутствие или отсутствие аудиопотоков в качестве индикатора. Состоянием по
 40 умолчанию, когда на систему клиента подается питание, является то, что включены все аудиоканалы. Формат одного варианта осуществления Пакета включения прямых аудиоканалов показан на фиг.30. Как показано на фиг.30, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Маски включения аудиоканала и ЦИК. Этот тип пакета, как правило, идентифицируется как пакет
 45 Типа 78 в 1-байтовом поле типа и использует предварительно выбранную фиксированную длину, равную 4 байтам.

21. Пакеты частоты аудиоотсчетов обратной линии передачи данных

Этот тип пакета обеспечивает структуру, способ или средство, которые позволяют хосту
 50 включать или выключать аудиоканалы в клиенте. Эта возможность является полезной, так что клиент может отключить питание от усилителей звуковой частоты для экономии питания, когда нет аудио для вывода хостом. Это значительно труднее осуществить неявным образом, используя присутствие или отсутствие аудиопотоков. Состоянием по умолчанию, когда на систему клиента подается питание или когда она

подключается к хосту, является то, что включены все аудиоканалы. Аудиосистема, подключенная к хосту и клиенту, должна быть готова или иметь возможность вывода аудиосигналов предполагаемым или требуемым образом в пределах примерно 100 мс или менее после того, как клиент примет пакет Включения прямых аудиоканалов, по меньшей

5 мере, один из битов в поле Маски включения аудиоканала которого выполнил переход состояния или значения из нуля в единицу. Клиент указывает возможность ответа на Пакет включения прямых аудиоканалов, используя значение, установленное для бита 15 поля Возможностей аудиоканала Пакета Возможностей клиента.

Этот пакет позволяет хосту включать или выключать аудиоканал обратной линии

10 передачи данных и устанавливать частоту отсчетов аудиоданных этого потока. Хост выбирает частоту отсчетов, которая определяется как действительная в Пакете возможностей клиента. Если хост выбирает недействительную частоту отсчетов, тогда клиент не посылает аудиопоток на хост и соответствующая ошибка, значение ошибки или сигнал ошибки может посылаться на хост в Пакете отчета об ошибках клиента. Хост может

15 выключить аудиопоток обратной линии передачи данных посредством установки частоты отсчетов на значение 255. Состоянием по умолчанию, принимаемым тогда, когда на систему клиента первоначально подается питание или когда она подключается, является выключенный аудиопоток обратной линии передачи данных. Формат одного варианта осуществления для Пакета частоты аудиоотсчетов обратной линии передачи данных

20 показан на фиг.31. Как показано на фиг.31, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Частоты аудиоотсчетов, Резервированное 1 и ЦИК. Этот тип пакета, как правило, идентифицируется как пакет Типа 79 и использует предварительно выбранную фиксированную длину, равную 4 байтам.

22. Пакеты Служебных данных защиты цифрового содержимого

Этот пакет обеспечивает структуру, способ или средство, которые позволяют хосту и клиенту обмениваться сообщениями, относящимися к используемому способу защиты

цифрового содержимого. В настоящее время рассматриваются два типа защиты содержимого: защита цифрового содержимого передачи (ЗЦСП) или широкополосная система защиты цифрового содержимого (ШСЗЦС), причем зарезервировано место для

30 обозначений будущих альтернативных схем защиты. Используемый способ задается параметром Типа защиты содержимого в этом пакете. Формат варианта осуществления Пакета служебных данных защиты цифрового содержимого показан на фиг.32. Как показано на фиг.32, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД bClient, Типа защиты содержимого, Служебных сообщений защиты

35 содержимого и ЦИК. Этот тип пакета, как правило, идентифицируется как пакет Типа 80.

23. Пакеты разрешения прозрачного цвета

Пакет разрешения прозрачного цвета представляет собой структуру, способ или средство, которые используются для задания того, какой цвет является прозрачным на

40 дисплее, и для разрешения или запрещения использования прозрачного цвета для отображения изображений. Дисплеи, которые обладают этой возможностью, сообщают об этой возможности битом 4 поля Возможностей свойств клиента Пакета возможностей клиента. Когда пиксель со значением для прозрачного цвета записывается в растровое изображение, то цвет не меняется с предыдущего значения. Формат Пакета разрешение прозрачного цвета показан на фиг.33. Как показано на фиг.33, в одном варианте

45 осуществления данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Разрешения прозрачного цвета, Резервированное 1, Идентификатора альфа-курсора, Дескриптора формата данных, Значения прозрачного пикселя и ЦИК. Этот тип пакета, как правило, идентифицируется как пакет Типа 81 в 1-байтовом поле типа и использует предварительно выбранную фиксированную длину, равную 10 байтам.

24. Пакеты измерения задержки на двойное прохождение

Пакет измерения задержки на двойное прохождение обеспечивает структуру, способ или средство, которые используются для измерения задержки на распространение от хоста до

50 клиента (дисплея) плюс задержка от клиента (дисплея) обратно до хоста. Это измерение

в действительности включает в себя задержки, которые существуют в линейных драйверах и приемниках и подсистеме межсоединений. Это измерение используется для установления параметров задержки на реверсирование передачи и делителя скорости передачи обратной линии передачи данных в Пакете инкапсуляция обратной линии

5 передачи данных, описанном в основном выше. Этот пакет наиболее полезен тогда, когда линия передачи данных ЦИМД работает с максимальной скоростью, предназначенной для конкретного применения. Пакет может посылаться в режиме Типа 1 и с меньшей скоростью передачи данных, чтобы увеличить диапазон измерения задержки на двойное прохождение. Сигнал MDDI_Stb ведет себя так, как будто посылаются все нулевые данные

10 во время следующих полей: обоих полей Защитных интервалов, Всех нулей и Периода измерения. Это вызывает переключение сигнала MDDI_Stb с частотой, равной половине скорости передачи данных, так что он может использоваться в качестве периодических тактовых импульсов в клиенте во время Периода измерения.

В одном варианте осуществления клиент, как правило, указывает возможность

15 поддержки Пакета измерения задержки на двойное прохождение посредством использования бита 18 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента. Рекомендуется, чтобы все клиенты поддерживали измерение задержки на двойное прохождение, но можно, чтобы хост имел сведения о задержке на двойное прохождение наихудшего случая, основываясь на максимальной задержке в

20 кабеле и на максимальных задержках драйвера и приемника. Хост также может заранее иметь сведения о задержке на двойное прохождение для линии передачи данных ЦИМД, используемой во внутреннем режиме, так как это представляет собой аспект известных элементов разработки (длины проводников, тип схемы и свойств и т.п.) устройства, в котором используется интерфейс.

25 Формат Пакета измерения задержки на двойное прохождение показан на фиг.34. Как показано на фиг.34, в одном варианте осуществления данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, ЦИК параметров, Защитного интервала 1, Периода измерения, Всех нулей и Защитного интервала 2. Этот тип пакета, как правило, идентифицируется как пакет Типа 82 и использует предварительно выбранную

30 фиксированную длину, равную 159 битам.

Временные соотношения событий, которые имеют место во время Пакета измерения задержки на двойное прохождение, изображены на фиг.35. На фиг.35 хост передает Пакет измерения задержки на двойное прохождение, показанный присутствием полей ЦИК параметров и Выравнивания строка, за которыми следуют поля Всех нулей 1 и Защитного

35 интервала 1. Задержка 3502 происходит до того, как пакет поступит на устройство дисплея клиента или обрабатывающую схему. Когда клиент принимает пакет, он передает комбинацию 0xff, 0xff и 30 байтов 0x00 практически насколько возможно точно в начале Периода измерения, что определяется клиентом. Фактическое время, когда клиент начинает передавать эту последовательность, задерживается от начала Периода

40 измерения по усмотрению хоста. Величина этой задержки по существу представляет собой время, затрачиваемое пакетом на распространение через линейные драйверы и приемники и подсистему межсоединений (кабели, проводники). Аналогичная величина задержки 3504 происходит при распространении комбинации от дисплея обратно на хост.

Для того чтобы точно определить время задержки на двойное прохождение для

45 сигналов, проходящих на клиент и от него, хост подсчитывает количество периодов битовых интервалов прямой линии передачи данных, происходящих после начала поля Периода измерения до тех пор, пока не будет обнаружено появление начала последовательности 0xff, 0xff и 30 байтов 0x00. Эта информация используется для определения величины времени распространения сигнала двойного прохождения от хоста

50 до клиента и снова обратно. Затем примерно половина этой величины приписывается задержке, создаваемой для одностороннего прохождения сигнала до клиента.

Как хост, так и клиент оба возбуждают линию уровнем логического нуля во время обоих защитных интервалов для поддержания линий MDDI_DATA в определенном состоянии.

Времена включения и выключения хоста и клиента во время обоих защитных интервалов такие, что сигналы MDDI_Data находятся на действительном низком уровне для любого действительного времени задержки на двойное прохождение.

25. Пакет калибровки перекося прямой линии передачи данных

5 Пакет калибровки перекося прямой линии передачи данных дает возможность клиенту или дисплею откалибровать самого себя вследствие различий в задержке на распространение сигналов MDDI_Data относительно сигнала MDDI_Stb. Без компенсации перекося задержки максимальная скорость передачи данных, как правило, ограничивается учетом потенциального разброса наихудшего случая этих задержек. Как правило, этот
10 пакет посылается только тогда, когда скорость передачи данных прямой линии передачи данных конфигурируется на скорость передачи примерно 50 Мбит/с или ниже. После посылки этого пакета для калибровки дисплея скорость передачи данных может ступенчато повышаться свыше 50 Мбит/с. Если скорость передачи данных устанавливается слишком
15 высокой во время процесса калибровки перекося, дисплей может синхронизироваться с паразитным сигналом периода бита, что может вызвать установку компенсации перекося задержки в выключенное состояние в продолжении более чем одного битового интервала, приводя к ошибочному тактированию данных. Наибольший тип скорости передачи данных интерфейса или наибольший возможный Тип интерфейса выбирается до посылки Пакета калибровки перекося прямой линии передачи данных, так что калибруются все
20 существующие биты данных.

Один вариант осуществления формата Пакета калибровки перекося прямой линии передачи данных показан на фиг.56. Как показано на фиг.56, данный тип пакета структурирован так, что имеет поля Длины пакета (2 байта), Типа пакета, ИД hClient, ЦИК параметров, Всех нулей 1, Последовательности данных калибровки и Всех нулей 2.
25 Этот тип пакета, как правило, идентифицируется как пакет Типа 83 в поле типа, и в одном варианте осуществления имеет предварительно выбранную длину, равную 519.

Виртуальная панель управления

Использование Виртуальной панели управления (ВПУ) дает возможность хосту устанавливать определенные элементы пользовательского управления в клиенте.

30 Посредством предоставления возможности регулировки этих параметров хостом, пользовательский интерфейс в клиенте может быть упрощен, так как экраны, которые дают возможность пользователю регулировать параметры, такие как громкость звука или яркость дисплея, могут генерироваться программным обеспечением хоста, а не одним или несколькими микропроцессорами в клиенте. Хост обладает возможностью считывания
35 установок параметров в клиенте и определения диапазона действительных значений для каждого элемента управления. Клиент, как правило, имеет возможность сообщить обратно хосту, какие параметры управления могут регулироваться.

Коды управления (Коды ВПУ) и связанные с ними значения данных, в основном заданные, используются для задания элементов управления и установок в клиенте. Коды
40 ВПУ в спецификации ЦИМД расширяются до 16 битов для сохранения надлежащего выравнивания полей данных в определениях пакетов, и в будущем для поддержки дополнительных значений, которые являются уникальными для данного интерфейса или будущих усовершенствований.

26. Пакет запроса свойств ВПУ

45 Пакет запроса свойств ВПУ обеспечивает средство, механизм или способ для запроса хостом текущих установок заданного параметра управления или всех действительных параметров управления. Как правило, клиент отвечает на Пакет ВПУ соответствующей информацией в Пакете ответа свойств ВПУ. В одном варианте осуществления клиент указывает возможность поддержки Пакета запроса свойств ВПУ, используя бит 13 поля
50 Индикаторов возможностей свойств клиента Пакета возможностей клиента.

Формат Пакета запроса свойств ВПУ в одном варианте осуществления показан на фиг.69. Как показано на фиг.69, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, кода ВПУ набора команд управления монитором

(НКУМ) и ЦИК. Этот тип пакета, как правило, идентифицируется в одном варианте осуществления как Тип 128, который указывается в 2-байтовом поле типа. Длина пакета, которая задает общее количество байтов в пакете, не включая поле длины пакета, обычно фиксирована для этого типа пакета с длиной 8 байтов.

- 5 Поле ИД hClient резервируется для использования в качестве ИД клиента в будущих реализациях и обычно устанавливается в нуль. Поле Кода ВПУ НКУМ содержит 2 байта информации, которая задает Параметр кода управления ВПУ НКУМ. Значение в диапазоне от 0 до 255 вызывает возврат Пакета ответа свойств ВПУ с единственным элементом в
- 10 Списке ответа свойств ВПУ, соответствующем заданному коду НКУМ. Код ВПУ НКУМ 65535 (0xffff) запрашивает Пакет ответа свойств ВПУ со Списком ответа свойств ВПУ, содержащим Элемент списка ответа свойств для каждого элемента управления, поддерживаемого клиентом. Значения 256-65534 для этого поля резервируются для будущего использования и в настоящее время не используются.

27. Пакет ответа свойств ВПУ

- 15 Пакет ответа свойств ВПУ обеспечивает средство, механизм или способ для ответа клиентом на запрос хоста с текущей установкой заданного параметра управления или всех действительных параметров управления. Как правило, клиент посылает Пакет ответа свойств ВПУ в ответ на Пакет запроса свойств ВПУ. Этот пакет является полезным для определения текущей установки заданного параметра, для определения действительного
- 20 диапазона для заданного элемента управления, для определения, поддерживается ли клиентом заданный элемент управления, или для определения набора элементов управления, которые поддерживаются клиентом. Если посылается Запрос свойств ВПУ, который ссылается на заданный элемент управления, который не реализован в клиенте, тогда Пакет ответа свойств ВПУ возвращается с единственным элементом Списка ответа
- 25 свойств ВПУ, соответствующим нереализованному элементу управления, который содержит соответствующий код ошибки. В одном варианте осуществления клиент указывает возможность поддержки Пакета ответа свойств ВПУ, используя бит 13 поля Возможностей свойств клиента Пакета возможностей клиента.

Формат Пакета ответа свойств ВПУ в одном варианте осуществления показан на фиг.70.

- 30 Как показано на фиг.70, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Версии НКУМ, Порядкового номера ответа, Списка ответа свойств ВПУ и ЦИК. Этот тип пакета, как правило, идентифицируется в одном варианте осуществления как Тип 129, как указано в 2-байтовом поле типа.

- 35 Поле ИД cClient содержит информацию, зарезервированную для ИД клиента. Это поле резервируется для будущего использования и, как правило, устанавливается в нуль. Поле Версии НКУМ содержит 2 байта информации, которая задает Версию Спецификации НКУМ АСВЭ, реализованной клиентом.

- 2-байтовое поле Порядкового номера ответа содержит информацию или данные, которые задают порядковый номер Пакетов ответа свойств ВПУ, возвращаемых клиентом.
- 40 Клиент возвращает один или несколько Пакетов ответа свойств ВПУ в ответ на Пакет запроса свойств ВПУ со значением Кода управления НКУМ, равным 65535. Клиент может расширить или пересылать список ответа свойств по многочисленным Пакетам ответа свойств ВПУ. В данном случае клиент должен назначить порядковый номер или идентификатор каждому последовательному пакету, и порядковые номера Пакетов ответа
- 45 свойств ВПУ, посылаемых в ответ на единственный Пакет запроса свойств ВПУ, обычно начинаются с нуля и увеличиваются на единицу. Последний Элемент списка ответа свойств ВПУ в последнем Пакете ответа свойств ВПУ должен содержать значение Кода управления ВПУ НКУМ, равное 0xffff, чтобы идентифицировать, что пакет является последним и содержит наивысший порядковый номер из группы возвращаемых пакетов.
- 50 Если только один Пакет ответа свойств ВПУ посылается в ответ на Пакет запроса свойств ВПУ, тогда Порядковый номер ответа в этом единственном пакете, как правило, устанавливается в нуль, и Список ответа свойств ВПУ содержит элемент списка, имеющий Код ВПУ НКУМ в Элементе Списка ответа свойств ВПУ, равный 0xffff. Поля Максимального

значения и Текущего значения (фиг.71) в пакете Элемента списка ответа свойств ВПУ устанавливаются в нуль, когда Код управления ВПУ НКУМ равен 0xffff.

Поле Количества свойств в списке содержит 2 байта, которые задают количество Элементов списка ответа свойств ВПУ, которые находятся в Списке ответа свойств ВПУ в этом пакете, тогда как поле Списка ответа свойств ВПУ представляет собой группу байтов, которые содержат один или несколько Элементов списка ответа свойств ВПУ. Формат единственного Элемента списка ответа свойств ВПУ в одном варианте осуществления показан на фиг.71.

Как показано на фиг.71, каждый Элемент списка ответа свойств ВПУ имеет длину 12 байтов и содержит поля Кода ВПУ НКУМ, Кода результата, Максимального значения и Текущего значения. 2-байтовое поле Кода ВПУ НКУМ содержит данные или информацию, которая задает Параметр кода управления ВПУ НКУМ, связанный с этим элементом списка. Только значения Кода управления, определенные в Спецификации НКУМ АСВЭ версии 2 и более поздней, считаются действительными для данного варианта осуществления. 2-байтовое поле Кода результата содержит информацию, которая задает код ошибки, относящийся к запросу информации, касающейся заданного Элемента управления ВПУ НКУМ. Значение «0» в данном поле означает, что нет ошибки, тогда как значение «1» означает, что заданный элемент управления не реализован в клиенте. Дополнительные значения для этого поля 2-65535 в настоящее время зарезервированы для будущего использования и реализации другого применения, рассматриваемого в данной области техники, но не должны использоваться в настоящее время.

4-байтовое поле Максимального значения задает наибольшее возможное значение, на которое может быть установлен заданный Элемент управления НКУМ. Если запрашиваемый элемент управления не реализован в клиенте, это значение устанавливается в нуль. Если длина возвращаемого значения меньше 32 битов (4 байтов), тогда значение приводится в 32-битовое целое число, оставляя самые старшие (неиспользуемые) байты, установленными в нуль. 4-байтовое поле Текущего значения содержит информацию, которая задает текущее значение задаваемого элемента непрерывного (Н) или прерывистого (П) управления ВПУ НКУМ. Если запрашиваемый элемент управления не реализован в клиенте или если элемент управления реализован, но представляет собой тип табличных (Т) данных, тогда это значение устанавливается в нуль. Если длина возвращаемого значения меньше 32 битов (4 байтов) согласно спецификации НКУМ АСВЭ, тогда значение приводится к 32-битовому целому числу, оставляя самые старшие (неиспользуемые) байты, установленными в нуль. Если задаваемый код ВПУ НКУМ соответствует элементу прерывистого управления или типу табличных данных, тогда поле Максимального значения устанавливается или выбирается равным нулю.

28. Пакет установки свойств ВПУ

Пакет установки свойств ВПУ обеспечивает средство, механизм или способ для установления хостом значений элемента управления ВПУ для как непрерывного, так и прерывистого управления в клиенте. В одном варианте осуществления клиент указывает возможность поддержки Пакета установки свойств ВПУ, используя бит 13 поля Возможностей свойств клиента Пакета возможностей клиента.

Формат Пакета установки свойств ВПУ в одном варианте осуществления показан на фиг.72. Как показано на фиг.72, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Кода ВПУ НКУМ, Количества значений в списке, Списка значений элемента управления и ЦИК. Этот тип пакета, как правило, идентифицируется как Тип 130, что указывается в 2-байтовом поле типа, имеет длину 20 байтов, исключая поле Длины пакета.

Поле ИД hClient снова использует 2-байтовое значение для задания или действия в качестве ИД клиента. Это поле резервируется для будущего использования и в настоящее время устанавливается в нуль. Поле Кода ВПУ НКУМ использует 2 байта информации или значений для задания Параметра кода управления ВПУ НКУМ, подлежащего регулировке.

2-байтовое Поле количества значений в списке содержит информацию или значения, которые задают количество 16-битовых значений, которые существуют в Списке значений элемента управления. Список значений элемента управления обычно содержит один элемент, если только Код управления НКУМ не относится к таблице в клиенте. В случае не относящихся к таблице элементов управления Список значений элементов управления содержит значение, которое задает новое значение, подлежащее записи в параметр управления, заданный полем Кода ВПУ НКУМ. Для относящихся к таблице элементов управлений формат данных в Списке значений элементов управления задается описанием параметра заданного кода ВПУ НКУМ. Если список содержит значения, которые больше одного байта, тогда самый младший байт передается первым, единообразно со способом, определенным в другом месте. Наконец, 2-байтовое поле ЦИК содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

29. Пакет запроса действительных параметров

Пакет запроса действительных параметров используется в качестве средства или структуры, полезной для запроса, чтобы клиент возвращал Пакет ответа действительных параметров, содержащий список параметров, поддерживаемых заданным элементом прерывистого (П) или табличного (Т) управлением. Этот пакет должен задавать только прерывистые управления или управления, которые относятся к таблице в клиенте, и не задавать значение Кода ВПУ НКУМ, равное 65535 (0xffff), для задания всех элементов управлений. Если задается неподдерживаемый или недействительный Код ВПУ НКУМ, тогда возвращается соответствующее значение ошибки в Пакете ответа действительных параметров. В одном варианте осуществления клиент указывает возможность поддержки Пакета запроса действительных параметров, используя бит 13 поля Возможностей свойств клиента Пакета возможностей дисплея.

Формат Пакета запроса действительных параметров в одном варианте осуществления показан на фиг.73. Как показано на фиг.73, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, Кода ВПУ НКУМ и ЦИК. Этот тип пакета, как правило, идентифицируется в одном варианте осуществления как Тип 131, что указывается в 2-байтовом поле типа.

Длина пакета, как указывается в 2-байтовом Поле длины пакета, как правило, устанавливается так, что имеет общее количество байтов в пакете, не включая поле длины пакета, равное 8. ИД hClient снова задает ИД клиента, но в настоящее время резервируется для будущего использования, что очевидно для специалиста в данной области техники, и устанавливается в нуль. 2-байтовое Поле кода ВПУ НКУМ содержит значение, которое задает Параметр кода прерывистого управления ВПУ НКУМ, подлежащий запросу. Значение в этом поле должно соответствовать прерывистому управлению, которое реализовано в клиенте. Значения 256-65535 (0xffff) обычно резервируются или рассматриваются как недействительные, и рассматриваются как нереализованное управление в ответе об ошибке.

30. Пакет ответа действительных параметров

Пакет ответа действительных параметров посылается в ответ на Пакет запроса действительных параметров. Он используется в качестве средства, способа или структуры для идентификации действительных установок для прерывистого управления ВПУ НКУМ или управления, которое возвращает содержимое таблицы. Если управление относится к таблице в клиенте, тогда Список ответа параметров ВПУ просто содержит заданный список последовательных значений таблицы, которые были запрошены. Если содержимое таблицы не может уместиться в одном Пакете ответа действительных параметров, тогда клиентом могут посылаться многочисленные пакеты с последовательными Порядковыми номерами ответов. В одном варианте осуществления клиент указывает возможность поддержки Пакета ответа действительных параметров, используя бит 13 поля Возможностей свойств клиента Пакета возможностей клиента.

Хост может запросить содержимое таблицы следующим образом: хост посылает Пакет установки свойств ВПУ, содержащий необходимые или требуемые параметры, такие как

параметр считывания/записи, смещения Таблицы преобразования (ТП) и выбор формата RGB; затем хостом посылается Пакет запроса действительных параметров, который задает требуемое управление; затем клиент возвращает один или несколько Пакетов ответа действительных параметров, содержащих табличные данные. Эта последовательность операций выполняет подобную функцию, что и функции считывания таблицы, описанные в модели работы НКУМ.

Если заданный параметр клиента не поддерживается клиентом, тогда в одном варианте осуществления соответствующее поле в данном пакете будет содержать значение 255. Для параметров, которые используются в клиенте, соответствующее поле должно содержать значение параметра в клиенте.

Формат Пакета ответа действительных параметров для одного варианта осуществления показан на фиг.74. Как показано на фиг.74, данный тип пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Кода ВПУ НКУМ, Кода ответа, Порядкового номера ответа, Количества значений в списке, Списка ответов параметров ВПУ и ЦИК. Этот тип пакета, как правило, идентифицируется для одного варианта осуществления как Тип 132, что указывается в 2-байтовом поле типа.

Поле ИД cClient резервируется для будущего ИД клиента, что известно из вышеупомянутых обсуждений, тогда как 3-байтовый Пакет кода ВПУ НКУМ содержит значение, которое задает Параметр кода прерывистого управления ВПУ НКУМ, который описывается этим пакетом. Если недействительный Код управления ВПУ НКУМ задается Пакетом запроса действительных параметров, тогда это же значение недействительного параметра будет задаваться в этом поле с соответствующим значением в поле Кода ответа. Если Код управления НКУМ является недействительным, тогда Список ответа параметров ВПУ будет иметь нулевую длину.

Поле Кода ответа содержит 2 байта информации или значений, которые задают сущность ответа, относящегося к запросу информации, касающейся заданного управления ВПУ НКУМ. Если значение в этом поле равно 0, тогда считается, что ошибка не присутствует для этого типа данных, и посылается последний Пакет ответа действительных параметров в последовательности, причем он имеет наивысший Порядковый номер ответа. Если значение в этом поле равно 1, тогда считается, что ошибка не присутствует, но другие Пакеты ответа действительных параметров будут посылаться, которые имеют более высокие порядковые номера. Если значение в этом поле равно 2, тогда заданное управление считается нереализованным в клиенте. Если значение в этом поле равно 3, тогда заданное управление не является прерывистым управлением (именно непрерывное управление всегда имеет действительный набор всех значений от нуля до его максимального значения). Значение для этого поля, равные 4-65535, резервируются для будущего использования и, как правило, не должны использоваться.

2-байтовое поле Порядкового номера ответа задает порядковый номер Пакетов ответа действительных параметров, возвращаемых клиентом. Клиент возвращает один или несколько Пакетов ответа действительных параметров в ответ на Пакет запроса действительных параметров. Клиент может расширить Список ответа параметров ВПУ по многочисленным Пакетам ответа действительных параметров. В этом последнем случае клиент назначает порядковый номер каждому последовательному пакету и устанавливает Код ответа в 1 во всех кроме последнего пакета в последовательности. Последний Пакет ответа действительных параметров в последовательности будет иметь наивысший Порядковый номер ответа, и Код ответа содержит значение 0.

2-байтовое поле Количества значений в списке задает количество 16-битовых значений, которые существуют в Списке ответа параметров ВПУ. Если Код ответа не равен нулю, тогда параметр Количества значений в списке равен нулю. Поле Списка ответа параметров ВПУ содержит список 0-32760 2-байтовых значений, которые указывают набор действительных значений для прерывистого управления, заданного полем Кода управления НКУМ. Определения кодов прерывистого управления задаются в Спецификации НКУМ АСВЭ. Наконец, в данном варианте осуществления поле ЦИК

содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

Изображения масштабируемых видеопотоков

ЦИМД или механизм, структура, средство или способ протокола обеспечивает поддержку изображений масштабируемых видеопотоков, которые дают возможность хосту
 5 посылать изображение клиенту, которое масштабируется больше или меньше, чем оригинальное изображение, и масштабированное изображение копируется в буфер основного изображения. Обзор функциональных возможностей Масштабируемого видеопотока и связанной с ним поддержки протокола предусмотрено в другом месте. Возможность поддержки масштабируемых видеопотоков определяется посредством или
 10 внутри Пакета возможностей масштабируемого видеопотока, который посылается в ответ на Пакет запроса заданного состояния.

Заголовок пакета Масштабируемого видеопотока, обсуждаемого ниже, несколько отличается от более простого Пакета видеопотока, заголовок которого содержит весь контекст, необходимый для отображения изображения. Пакет Масштабируемого
 15 видеопотока использует пакет установки для определения параметров размера и положения окна источника и назначения и отдельный Пакет масштабируемого видеопотока для передачи данных пикселей. Клиент распределяет внутреннее запоминающее устройство, связанное с каждым потоком, для хранения параметров потока из пакета установки и часть данных пикселей, связанных с каждым потоком. Емкость запоминающего
 20 устройства, необходимого для каждого потока, будет изменяться в зависимости от размера изображений источника и назначения и значений, заданных в пакете установки. По этой причине протокол разрабатывается так, чтобы дать возможность клиенту реализовать динамическое распределение памяти для назначения запоминающего устройства, связанного с каждым масштабируемым видеопотоком.

Полезно посылать видеопоток на дисплей, имеющий размер, присущий программному источнику, и чтобы дисплей масштабировал и располагал изображение таким образом, которое подходит для заданного конечного применения. Реализация масштабирования в реальном времени многочисленных видеоизображений достаточно сложная, чтобы
 25 осуществить дополнительно поддержку этого свойства в клиенте.

31. Пакет возможностей масштабируемого видеопотока

Пакет возможностей масштабируемого видеопотока определяет характеристики изображения источника масштабируемого видеопотока в клиенте или используемого клиентом. Формат Пакета возможностей масштабируемого видеопотока показан в основном на фиг.75. Как показано на фиг.75, в одном варианте осуществления Пакет
 35 возможностей масштабируемого видеопотока структурирован так, что имеет поля Длины пакета, Типа пакета ИД cClient, Максимального количества потоков, Максимального размера источника по оси X, Максимального размера источника по оси Y, Возможностей формата RGB, Возможностей монохромного формата, Резервированное 1, Возможностей формата Y Cr Cb, Резервированное 2 и ЦИК. Длина пакета в одном варианте
 40 осуществления выбирается равной фиксированным 20 байтам, как показано в поле длины, включая 2-байтовое поле ИД cClient, которое резервируется для использования для ИД клиента, иначе устанавливается в нуль, и поле ЦИК. В одном варианте осуществления клиент указывает возможность поддержки Пакета возможностей масштабируемого видеопотока, используя значение параметра 143 в Списке ответа действительных
 45 параметров Пакета списка ответа действительных состояний.

2-байтовое поле Максимального количества потоков содержит значения для идентификации максимального количества одновременно масштабируемых видеопотоков, которые могут распределяться одновременно. В одном варианте осуществления клиент должен отклонить запрос на выделение масштабируемого видеопотока, если уже выделено
 50 максимальное количество масштабируемых видеопотоков. Если выделено меньше максимального количества масштабируемых видеопотоков, клиент также может отклонить запрос на выделение, основываясь на других ограничениях ресурсов в клиенте.

Поля (2 байта) Максимального размера источника по оси X и размера источника по оси

У задают значения для максимальной ширины и высоты соответственно изображения источника масштабируемого видеопотока, выраженного в виде количества пикселей.

Поле Возможностей формата RGB использует значения для задания числа битов разрешения, с которым может отображаться в формате RGB. Если масштабируемый видеопоток не может использовать формат RGB, тогда это значение устанавливается равным нулю. Слово Возможностей формата RGB состоит из трех отдельных значений без знака: Биты 3-0 определяют максимальное количество битов синего цвета (яркость синего цвета) в каждом пикселе, Биты 7-4 определяют максимальное количество битов зеленого цвета (яркость зеленого цвета) в каждом пикселе, и Биты 11-8 определяют максимальное количество битов красного цвета (яркость красного цвета) в каждом пикселе, тогда как Биты 15-12 резервируются для будущего использования при определении будущих возможностей и, как правило, устанавливаются в нуль.

1-байтовое поле Возможностей монохромного формата содержит значение, которое задает количество битов разрешения, которое может отображаться в монохромном формате. Если масштабируемый видеопоток не может использовать монохромный формат, тогда это значение устанавливается в нуль. Биты 7-4 резервируются для будущего использования и должны поэтому быть установлены в нуль («0») для текущих применений, хотя это может измениться со временем, что понятно для специалиста в данной области техники. Биты 3-0 определяют максимальное количество битов серой шкалы, которая может существовать в каждом пикселе. Эти четыре бита делают возможным задавать, чтобы каждый пиксель состоял из 1-15 битов. Если значение равно нулю, тогда монохромный формат не поддерживается масштабируемым видеопотоком.

Поле Резервированное 1 (в данном случае 1 байт) резервируется для будущего использования для предоставления значений, относящихся к информации или данным Пакета масштабируемого видеопотока. Поэтому в настоящее время все биты в этом поле устанавливаются на логический «0». Одним назначением этого поля является то, чтобы вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

2-байтовое поле Возможностей формата Y Cb Cr содержит значения, которые задают количество битов разрешения, которое может отображаться в формате Y Cb Cr. Если масштабируемый видеопоток не может использовать формат Y Cb Cr, тогда это значение равно нулю. Слово Возможностей формата Y Cb Cr состоит из трех отдельных значений без знака: Биты 3-0 определяют максимальное количество битов, которые задают отсчет Cr; Биты 7-4 определяют максимальное количество битов, которые задают отсчет Cb; Биты 11-8 определяют максимальное количество битов, которые задают отсчет Y; и Биты 15-12 резервируются для будущего использования и, как правило, устанавливаются в нуль.

1-байтовое поле Битов возможностей содержит набор флагов, которые задают возможности, связанные с масштабируемым видеопотоком. Флаги определяются следующим образом: Бит 0 охватывает данные Пикселей в Пакете масштабируемого видеопотока, которые могут быть в упакованном формате. Пример упакованных и выровненных по байтам данных пикселей показан ранее на фиг.12. Бит 1 резервируется для будущего использования и, как правило, устанавливается в нуль; Бит 2 также резервируется для будущего использования и устанавливается в нуль; Бит 3 охватывает масштабируемые видеопотоки, которые могут задаваться в формате данных карты цветов. Эта же таблица карты цветов используется для масштабируемых видеопотоков, что и используется для буфера основных изображений и плоскостей изображения альфа-курсора. Карта цветов конфигурируется с использованием Пакета карты цветов, описанного в другом месте; и Биты 7-4 резервируются для будущего использования и, как правило, устанавливаются равными нулю.

Поле (в данном случае 1 байт) Резервированное 2 резервируется для будущего использования при предоставлении значений, относящихся к информации или данным Пакета масштабируемого видеопотока. Поэтому в настоящее время все биты в этом поле устанавливаются на логический «0». Одним назначением этого поля является то, чтобы

вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

32. Пакет установки масштабируемого видеопотока

Пакет установки масштабируемого видеопотока обеспечивает средство, структуру или способ, используемый для определения параметров масштабируемого видеопотока, и клиент использует информацию для распределения внутреннего запоминающего устройства для буферизации и масштабирования изображения. Поток может быть освобожден посредством посылки этого пакета с полями Размера изображения по оси X и Размера изображения по оси Y равными нулю. Масштабируемые видеопотоки, которые были освобождены, могут быть снова выделены позже с этими же или другими параметрами потока. В одном варианте осуществления клиент указывает возможность поддержки Пакета установки масштабируемого видеопотока, используя значение параметра 143 в Списке ответа действительных параметров Пакета списка ответа действительных состояний и посредством использования ненулевого значения в поле Максимального количества потоков Пакета возможностей масштабируемого видеопотока.

Формат Пакета установки масштабируемого видеопотока показан в основном на фиг.76. Как показано на фиг.76, в одном варианте осуществления Пакет установки масштабируемого видеопотока структурирован так, что имеет поля Длины пакета, Типа пакета, hClient, ИД потока, Дескриптора формата визуальных данных, Атрибутов данных пикселей, Левого края по оси X, Верхнего края по оси Y, Правого края по оси X, Нижнего края по оси Y, размера изображения по оси X, Размера изображения по оси Y и ЦИК.

2-байтовое поле Длины пакета задает общее количество байтов в пакете, не включая поле длины пакета. В одном варианте осуществления длина этого пакета является фиксированной на значении 24. 2-байтовое поле Типа Пакета использует значение 136 для идентификации пакета в качестве Пакета установки масштабируемого видеопотока. 2-байтовое поле ИД hClient резервируется для будущего использования в качестве ИД клиента и, как правило, устанавливается всеми битами на значение логического нуля для настоящего момента или до тех пор, пока пользователь протокола не определит, что значения ИД должны использоваться, что будет известным.

Поле ИД потока использует 2 байта для задания уникального идентификатора для ИД потока. Это значение назначается хостом и находится в диапазоне значений от нуля до максимального значения ИД потока, заданного в Пакете возможностей клиента. Хост должен тщательно управлять использованием значений ИД потока, чтобы гарантировать, чтобы каждому активному потоку назначалось уникальное значение и чтобы потоки, которые больше не являются активными, освобождались или переназначались.

В одном варианте осуществления поле Дескриптора формата видеоданных использует 2 байта для задания формата каждого пикселя в Данных пикселя в текущем потоке в текущем пакете. Формат данных пикселей должен соответствовать по меньшей мере одному из действительных форматов для плоскости изображения альфа-курсора, как могло быть определено в Пакете возможностей изображения альфа-курсора, или другом предварительно определенном узоре изображения, как в основном будет определено в других пакетах, обсужденных выше. Дескриптор формата видеоданных определяет формат пикселя только для текущего пакета и не означает, что постоянный формат будет продолжаться использоваться в течение времени существования конкретного видеопотока. Фиг.11 иллюстрирует вариант осуществления того, как кодируется Дескриптор формата видеоданных и как описано выше для других пакетов.

В одном варианте осуществления 2-байтовое поле Атрибутов данных пикселей имеет значения, которые интерпретируются следующим образом, при этом Биты 1 и 0 резервируются для будущего использования, как правило, в настоящее время устанавливаются в нуль, и Бит 2 указывает, находятся ли или нет Данные пикселей в чересстрочном формате. Когда Бит 2 равен 0, тогда Данные пикселей находятся в стандартном последовательном формате. Номер строки (координата Y пикселя) получает

приращение на 1 при продвижении с одной строки на следующую. Когда Бит 2 равен 1, тогда Данные пикселей находятся в чересстрочном формате. Номер строки (координата Y пикселя) получает приращение на 2 при продвижении с одной строки на следующую.

Бит 3 указывает, находятся ли или нет Данные пикселей в формате чередующихся пикселей. Это аналогично стандартному чересстрочному режиму, включаемому битом 2, но чередование является вертикальным вместо горизонтального. Когда Бит 3 равен 0, Данные пикселей находятся в стандартном последовательном формате. Номер столбца (координата X пикселя) получает приращение на 1 по мере приема каждого последующего пикселя. Когда Бит 3 равен 1, тогда Данные пикселей находятся в формате чередующихся пикселей. Номер столбца (координата X пикселя) получает приращение на 2 по мере приема каждого пикселя.

Биты 4-15 также резервируются для будущего использования и, как правило, устанавливаются на уровень или значения логического нуля для настоящих применений или разработок.

33. *Пакет подтверждения масштабируемого видеопотока*

Пакет подтверждения масштабируемого видеопотока дает возможность клиенту подтвердить прием Пакета установки масштабируемого видеопотока. Клиент может указывать возможность поддержки Пакета подтверждения масштабируемого видеопотока через значение параметра 143 в Списке ответа действительных параметров Пакета списка ответа действительных состояний и через ненулевое значение в поле Максимального количества потоков Пакета возможностей масштабируемого видеопотока.

Формат Пакета подтверждения масштабируемого видеопотока показан в основном на фиг.77. Как показано на фиг.77, в одном варианте осуществления Пакет подтверждения масштабируемого видеопотока структурирован так, что имеет поля Длины пакета, Типа пакета, cClient, ИД потока, Кода подтверждения и ЦИК. 2-байтовое поле Длины пакета используется для задания общего количества байтов, исключая поле длины пакета, значением 10 для этого типа пакета, тогда как Тип пакета 137 идентифицирует пакет в качестве Пакета подтверждения масштабируемого видеопотока.

2-байтовое поле ИД cClient резервируется для будущего использования для ИД клиента и, как правило, устанавливается в нуль. 2-байтовое поле ИД потока задает уникальный идентификатор для ИД потока. Он представляет собой то же значение, что и назначенное хостом в Пакете установки масштабируемого видеопотока.

2-байтовое поле Кода подтверждения предоставляет значения, содержащие код, который описывает результат попытки обновления заданного масштабируемого видеопотока. В одном варианте осуществления коды определяются следующим образом:

- 0 - Попытка выделения потока была успешной.
 - 1 - Попытка освобождения потока была успешной.
 - 2 - Недействительная попытка выделения ИД потока, который уже был выделен.
 - 3 - Недействительная попытка освобождения ИД потока, который уже освобожден.
 - 4 - Клиент не поддерживает масштабируемые видеопотоки.
 - 5 - Параметры потока не согласуются с возможностями клиента.
 - 6 - Значение ИД потока больше максимального значения, разрешенного клиентом.
 - 7 - Недостаточные ресурсы, доступные в клиенте, для выделения заданного потока.
- 2-байтовое поле ЦИК содержит ЦИК всех байтов в пакете, включая Длину пакета.

34. *Пакет масштабируемого видеопотока*

Пакет масштабируемого видеопотока используется для передачи данных пикселей, связанных с заданным масштабируемым видеопотоком. Размер области, на которую ссылается этот пакет, определяется Пакетом установки масштабируемого видеопотока. Клиент может указывать возможность поддержки Пакета масштабируемого видеопотока посредством значения параметра 143 в Списке ответа действительных параметров Пакета списка ответа действительных состояний и с использованием ответа об успешном выделении масштабируемого видеопотока в поле Кода подтверждения Пакета подтверждения масштабируемого видеопотока.

Формат одного варианта осуществления Пакета масштабируемого видеопотока показан в основном на фиг.78. Как показано на фиг.78, Пакет масштабируемого видеопотока структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, ИД потока, Атрибутов данных пикселей, Числа пикселей, ЦИК параметров, Данных пикселей и ЦИК

5 данных пикселей. 2-байтовое поле Типа пакета использует значение 18 для идентификации пакета в качестве Пакета масштабируемого видеопотока. Поле ИД hClient резервируется для ИД клиента и, как правило, устанавливается в нуль. Как и раньше, 2-байтовое поле ИД потока задает уникальный идентификатор для ИД потока. Это значение задается хостом в Пакете установки масштабируемого видеопотока и подтверждается в Пакете
10 подтверждения масштабируемого видеопотока. В одном варианте осуществления 2-байтовое поле Атрибутов данных пикселей имеет значения, которые задают направление данных пикселей и обновление изображения дисплея или расположений буфера. Эти значения в одном варианте осуществления

В одном варианте осуществления 2-байтовое поле Атрибутов данных пикселей имеет
15 значения, которые задают направление данных пикселей и обновление изображения дисплея или расположений буфера. Эти значения в одном варианте осуществления интерпретируются следующим образом. Биты 1 и 0 выбирают дисплей, куда данные пикселей должны направляться. Для значений битов «11» или «00» данные пикселей отображаются для обоих глаз, для значений битов «10» данные пикселей направляются
20 только для левого глаза, и для значений битов «01» данные пикселей направляются только для правого глаза.

Биты 7 и 6 представляют собой Биты обновления изображения дисплея, которые задают буфер кадров, где должны быть записаны данные пикселей. Влияние Битов обновления кадра описываются более подробно в другом месте. Когда Биты [7:6] равны «01», данные
25 Пикселей записываются в автономный буфер изображения. Когда Биты [7:6] равны «00», данные Пикселей записываются в буфер изображения, используемый для обновления изображения дисплея. Когда Биты [7:6] равны «11», данные Пикселей записываются во все буферы изображения. Если Биты [7:6] равны «10», то это рассматривается как недействительное значение. Эти биты в настоящее время резервируются для будущего
30 использования. В данной ситуации данные Пикселей будут игнорироваться и не будут записываться ни в какой из буферов изображения. Биты 2-5 и 8-15 резервируются для будущего использования и, как правило, устанавливаются на уровень или значения логического нуля.

2-байтовое поле Числа пикселей задает количество пикселей в поле Данных пикселей
35 ниже. 2-байтовое поле ЦИК параметров имеет ЦИК всех байтов с Длины пакета до Числа пикселей. Если проверка этим ЦИК неуспешна, тогда весь пакет отбрасывается. 2-байтовое поле Данных пикселей содержит необработанную видеоинформацию, которая должна масштабироваться и затем отображаться. Данные форматируются так, как описывается полем Дескриптора формата видеоданных. Данные передаются построчно,
40 что определено ранее.

2-байтовое поле ЦИК данных пикселей содержит ЦИК только Данных пикселей. Если проверка этим ЦИК неуспешна, тогда Данные пикселей все же могут использоваться, но число ошибок ЦИК получает приращение.

35. Пакет запроса заданного состояния

45 Пакет запроса заданного состояния обеспечивает средство, механизм или способ для запроса хостом, чтобы клиент послал пакет возможностей или состояния обратно на хост, как задается в этом пакете. Клиент возвращает пакет заданного типа в следующем Пакете инкапсуляции обратной линии передачи данных. Клиент, как правило, устанавливает бит 17 в поле Возможностей свойств клиента Пакета возможностей клиента, если клиент имеет
50 возможность отвечать на Пакет запроса заданного состояния. Обычным способом использования хостом для определения всех типов пакетов состояния, которые клиент может вернуть или переслать, является использование Пакета списка ответа действительных состояний, описанного в другом месте. Клиент может указывать

возможность ответа при помощи Пакета списка ответа действительных состояний, используя бит 21 поля Возможностей свойств клиента Пакета возможностей клиента.

Формат одного варианта осуществления Пакета запроса заданного состояния показан в основном на фиг.79. Как показано на фиг.79, Пакет запроса заданного состояния

5 структурирован так, что имеет поля Длины пакета, Типа пакета, ИД hClient, ИД пакета состояния и ЦИК. Поле Длины пакета задает общее количество байтов в пакете, не включая поле длины пакета, и, как правило, фиксировано на значении 10 для этого типа пакета. Тип пакета 138 идентифицирует пакет в качестве Пакета запроса заданного состояния. Поле (2 байта) ИД hClient резервируется для будущего использования для ИД
10 клиента и устанавливается в настоящее время в нуль, тогда как 2-байтовое поле ИД пакета состояния задает тип пакета возможностей или состояния, который клиент собирается послать хосту. Типовыми типами пакетов являются:

66 - Пакет возможностей клиента посылается клиентом.

133 - Пакет возможностей изображения альфа-курсора посылается клиентом.

15 139 - Посылается пакет списка ответа действительных состояний, который идентифицирует точные типы пакетов возможностей и состояния, которые клиент может послать.

140 - Пакет параметров задержки на обработку пакета посылается клиентом.

141 - Пакет возможностей персонального клиента посылается клиентом.

20 142 - Пакет отчета об ошибках клиента посылается клиентом.

143 - Пакет возможностей масштабируемого видеопотока посылается клиентом.

144 - Пакет идентификации клиента посылается клиентом.

Типы 56-63 пакета могут использоваться для идентификаторов характерных для изготовителя возможностей и состояния.

25 Поле ЦИК снова содержит ЦИК всех байтов в пакете, включая Длину пакета.

36. Пакет списка ответа действительных состояний

Пакет списка ответа действительных состояний обеспечивает хосту структуру, средство или способ, чтобы он имел пакеты списка состояния и возможностей, которые клиент имеет возможность возвращения. Клиент может указывать возможность поддержки Пакета
30 списка ответа действительных состояний, используя бит 21 поля Возможностей свойств клиента Пакета возможностей клиента.

Формат одного варианта осуществления Пакета списка ответа действительных состояний показан в основном на фиг.80. Как показано на фиг.80, Пакет списка ответа действительных состояний структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Количества значений в списке, Списка ответа действительных
35 параметров и ЦИК. Длина пакета для этого типа пакета, как правило, фиксируется на значении 10, и значение типа 139 идентифицирует пакет в качестве Пакета ответа действительных состояний. Поле ИД cClient резервируется для будущего использования в качестве ИД клиента и, как правило, устанавливается в нуль. 2-байтовое поле
40 Количества значений в списке задает количество элементов в следующем Списке ответа действительных параметров.

Поле Списка ответа действительных параметров содержит список 2-байтовых параметров, которые задают типы пакетов возможностей или состояния, которые клиент может посылать хосту. Если клиент указал, что он может ответить на Пакет запроса заданного состояния (используя бит 21 поля Возможностей свойств клиента в Пакете
45 возможностей клиента), тогда он может посылать по меньшей мере Пакет возможностей клиента (Тип пакета = 66) и Пакет списка ответа действительных состояний (Тип пакета = 139). Типы пакета, которые могут посылаться клиентом и могут быть включены в этот список вместе с их соответствующими назначениями для целей одного варианта
50 осуществления, представляют собой:

66 - Пакет возможностей клиента.

133 - Пакет возможностей изображения альфа-курсора.

139 - Пакет списка ответа действительных состояний, который идентифицирует точные

типы пакетов возможностей и состояния, которые клиент может посылать.

140 - Пакет параметров задержки на обработку пакета.

141 - Пакет возможностей персонального дисплея.

142 - Пакет отчета об ошибках клиента.

5 143 - Пакет возможностей масштабируемого видеопотока.

144 - Пакет идентификации клиента.

145 - Пакет возможностей альтернативного дисплея.

Типы 56-63 пакета могут использоваться для идентификаторов характерных для изготовителя возможностей и состояния.

10 Поле ЦИК содержит ЦИК всех байтов в пакете, включая Длину пакета.

37. Пакет параметров задержки на обработку пакета

Пакет параметров задержки на обработку пакета обеспечивает набор параметров для того, чтобы дать возможность хосту вычислить время, необходимое для завершения обработки, связанной с приемом заданного типа пакета. Некоторые команды, посылаемые хостом, не могут быть завершены клиентом в начальный момент времени. Хост может опрашивать биты состояния в Пакете запроса клиента и состояния для определения, были ли завершены некоторые функции клиентом, или может ли хост вычислять время завершения, используя параметры, возвращаемые клиентом в Пакете параметров задержки на обработку пакета. Клиент может указывать возможность поддержки Пакета параметров задержки на обработку пакета, используя значение параметра 140 в Списке ответа действительных параметров Пакета списка ответа действительных состояний.

Формат одного варианта осуществления Пакета параметров задержки на обработку пакета показан в основном, на фиг.81А. Как показано на фиг.81А, Пакет параметров задержки на обработку пакета структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Количества элементов списка, Списка параметров задержки и ЦИК. Длина пакета для этого типа пакета, как правило, фиксирована на значение 10, и значение типа 140 идентифицирует пакет в качестве Пакета параметров задержки на обработку пакета. Поле ИД cClient резервируется для будущего использования в качестве ИД клиента и, как правило, устанавливается в нуль. 2-байтовое поле Количества элементов списка задает количество элементов в следующем Списке ответа действительных параметров.

Поле Списка параметров задержки представляет собой список, содержащий один или несколько элементов Списка параметров задержки. Формат для одного варианта осуществления одного элемента Списка параметров задержки показан на фиг.81В, где показаны поля Типа пакета для задержки, Задержки пикселей, Задержки горизонтальных пикселей, Задержки вертикальных пикселей и Фиксированной задержки.

Каждый Элемент списка параметров задержки в основном ограничивается длиной, равной 6 байтам, и дополнительно определяется следующим образом. 2-байтовое поле Типа пакета для задержки задает Тип пакета, для которого применяются следующие параметры задержки. Поле (1 байт) Задержки пикселей содержит индекс для значения задержки. Значение, считанное из таблицы, умножается на общее количество пикселей в поле назначения пакета. Общее количество пикселей равно ширине, умноженной на высоту области назначения растрового изображения, на которое ссылается пакет. 1-байтовое поле Задержки горизонтальных пикселей содержит значение, которое представляет собой индекс для таблицы значений задержки (такой же таблицы, что и цифровая линия пакетной передачи видеоданных (ЦЛППВД)). Значение, считанное из таблицы, умножается на ширину (в пикселях) поля назначения пакета. 1-байтовое поле Задержки вертикальных пикселей содержит значение, которое представляет собой индекс для таблицы значений задержки (использует в основном такую же таблицу, что и ЦЛППВД). Значение, считанное из таблицы, умножается на высоту (в пикселях) поля назначения пакета.

Поле Фиксированной задержки использует 1 байт в качестве индекса для таблицы значений задержки (такой же таблицы, что и ЦЛППВД). Значение, считанное из таблицы,

представляет собой параметр фиксированной задержки, который представляет время, необходимое для обработки пакета, которое не связано ни с какими значениями параметров, заданными в пакете. Общая задержка, или задержка на время завершения обработки пакета, определяется в соответствии с зависимостью:

$$\begin{aligned} \text{Задержка} = & (\text{PacketProcessingDelay}(\text{PixelDelay}).\text{TotalPixels}) + \\ & (\text{PacketProcessingDelay}(\text{horizontalPixelDelay}).\text{Width}) + \\ & (\text{PacketProcessingDelay}(\text{VerticalPixelDelay}).\text{Height}) + \\ & \text{PacketProcessingDelay}(\text{FixedDelay}) \end{aligned}$$

Для некоторых пакетов Total Pixels, Width или Height не применяются, так как на эти параметры нет ссылки в соответствующем пакете. В этих случаях соответствующий параметр Pixel Delay, как правило, устанавливается равным нулю.

38. Пакет возможностей персонального дисплея

Пакет возможностей персонального дисплея обеспечивает набор параметров, которые описывают возможности устройства персонального дисплея, такого как установленного на голове дисплея или дисплейные очки. Это дает возможность хосту настроить информацию дисплея в соответствии с заданными возможностями клиента. Клиент, с другой стороны, указывает возможность отправки Пакета возможностей персонального дисплея посредством использования соответствующего параметра в Списке ответа действительных параметров Пакета списка ответа действительных состояний.

Формат одного варианта осуществления Пакета возможностей персонального дисплея показан в основном на фиг.82. Как показано на фиг.82, Пакет возможностей персонального дисплея структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Схемы расположения субпикселей, Формы пикселей, Поля зрения по горизонтали, Поля зрения по вертикали, Пересечения зрительных осей, Левого/правого изображения, Сквозного просмотра, Максимальной яркости, Оптических возможностей, Минимального расстояния между зрачками (PM3), Максимального PM3, Списка точек кривизны поля и ЦИК. В одном варианте осуществления значения поля Длины пакета фиксирована на 68. Значение Типа пакета 141 идентифицирует пакет в качестве Пакета возможностей персонального дисплея. Поле ИД cClient резервируется для будущего использования и, как правило, в настоящее время устанавливается в нуль.

Поле Схемы расположения субпикселей задает физическую схему расположения субпикселей сверху вниз и слева направо, используя значения: 0 для указания того, что схема расположения субпикселей не определена; 1 для указания красной, зеленой, синей полосы; 2 для указания синей, зеленой, красной полосы; 3 для указания квадратного пикселя, имеющего размещение 2 на 2 субпикселя красного сверху слева, синего внизу справа и двух зеленых субпикселей, один внизу слева и другой вверху справа; 4 для указания квадратного пикселя с размещением 2 на 2 субпикселя красного внизу слева, синего сверху справа и двух зеленых субпикселей, одного сверху слева и другого внизу справа; 5 для указания Дельты (триады); 6 для указания мозаики с перекрывающимися красным, зеленым и синим цветами (например, дисплей на жидких кристаллах на кремнии (ЖКНК) с чередованием цветов по полям); и со значениями 7-255, как правило, резервируемыми для будущего использования.

Поле Формы пикселя задает форму каждого пикселя, который состоит из субпикселей заданной конфигурации, используя значение: 0 для указания того, что форма субпикселей не определена; 1 для указания круглой; 2 для указания квадратной; 3 для указания прямоугольной; 4 для указания овальной; 5 для указания эллиптической; и с значениями 6-255, зарезервированными для будущего использования при указании требуемых форм, что может быть понятно для специалиста в данной области техники.

1-байтовое поле Поля зрения по горизонтали (ПЗГ) задает поле зрения по горизонтали с шагом 0,5 градуса (например, если ПЗГ равно 30 градусов, это значение равно 60). Если это значение равно нулю, тогда ПЗГ не задано.

1-байтовое поле Поля зрения по вертикали (ПЗВ) задает поле зрения по вертикали с шагом 0,5 градуса (например, если ПЗВ равно 30 градусов, это значение равно 60). Если это значение равно нулю, тогда ПЗВ не задано.

1-байтовое поле пересечения зрительных осей задает пересечение зрительных осей с шагом 0,01 диоптрии (1/м) (например, если пересечение зрительных осей равно 2,22 метра, то это значение равно 45). Если это значение равно нулю, тогда Пересечение зрительных осей не задано.

5 1-байтовое поле Перекрытия левого/правого изображения задает процент перекрытия левого и правого изображения. Допустимый диапазон перекрытия изображений в процентах составляет от 1 до 100. Значения 101-255 являются недействительными и, как правило, не должны использоваться. Если это значение равно нулю, тогда перекрытие изображений не задано.

10 1-байтовое поле Сквозного просмотра задает процент сквозного просмотра изображения. Допустимый диапазон сквозного просмотра в процентах составляет от 0 до 100. Значения 101-254 являются недействительными и не должны использоваться. Если это значение равно 255, тогда процент сквозного просмотра не задан. 1-байтовое поле Максимальной яркости задает максимальную яркость с шагом 20 нит (например, если
15 максимальная яркость равна 100 нит, это значение равно 5). Если это значение равно нулю, тогда максимальная яркость не задана.

2-байтовое поле Флагов оптических возможностей содержит различные поля, которые задают оптические возможности дисплея. Эти битовые значения в основном назначаются в соответствии с:

20 Биты 15-5 резервируются для будущего использования и, как правило, устанавливаются в состояние логического нуля.

Бит 4 выбирает Регулировку фокусировки очков, причем значение «0» означает то, что дисплей не имеет регулировки фокусировки очков, и значение «1» означает то, что дисплей имеет регулировку фокусировки очков.

25 Биты 3-2 выбирают Бинокулярную функцию в соответствии с тем, что: значение 0 означает, что дисплей является бинокулярным и может отображать только 2-мерные (2D) изображения; 1 означает, что дисплей является бинокулярным и может отображать 3-мерные (3D) изображения; 2 означает, что дисплей является монокулярным, и 3 резервируется для будущего использования.

30 Биты 1-0 выбирают Симметрию кривизны левого-правого поля, причем значение 0 означает, что Кривизна поля не определена. Если это поле равно нулю, все значения кривизны поля от A1 до E5 устанавливаются в нуль за исключением точки C3, которая задает фокусное расстояние дисплея, или должна устанавливаться в нуль для указания того, что фокусное расстояние не задано. Значение 1 означает, что Левый и правый
35 дисплеи имеют одинаковую симметрию; 2 означает, что Левый и правый дисплеи зеркально отображаются относительно вертикальной оси (столбец C); и 3 резервируется для будущего использования.

1-байтовое поле Минимума расстояния между зрачками (PM3) задает минимальное расстояние между зрачками в миллиметрах (мм). Если это значение равно нулю, тогда
40 минимальное расстояние между зрачками не задано. 1-байтовое поле Максимума расстояния между зрачками (PM3) задает максимальное расстояние между зрачками в миллиметрах (мм). Если это значение равно нулю, тогда максимальное расстояние между зрачками не задано.

Поле Списка точек кривизны поля содержит список 25 2-байтовых параметров, которые
45 задают фокусное расстояние в тысячных долях диоптрии (1/м) с диапазоном от 1 до 65535 (например, 1 равна 0,001 диоптрии, и 65535 равно 65,535 диоптрии). 25 элементов в Списке точек кривизны поля обозначаются A1-E5, как показано на фиг.83. Точки должны равномерно распределяться по активной области дисплея. Столбец C соответствует вертикальной оси дисплея, и строка 3 соответствует горизонтальной оси дисплея.

50 Столбцы A и E соответствуют левому и правому краям дисплея соответственно. И строки 1 и 5 соответствуют верхнему и нижнему краям дисплея соответственно. Порядок 25 точек в списке следующий: A1, B1, C1, D1, E1, A2, B2, C2, D2, E2, A3, B3, C3, D3, E3, A4, B4, C4, D4, E4, A5, B5, C5, D5, E5.

Поле ЦИК содержит ЦИК всех байтов в пакете, включая Длину пакета.

39. Пакет отчета об ошибках клиента

Пакет отчета об ошибках клиента действует в качестве механизма или средства, позволяющего клиенту предоставлять хосту список рабочих ошибок. Клиент может обнаружить ошибки в широком диапазоне в ходе своей нормальной работы в результате приема некоторых команд от хоста. Примеры таких ошибок включают в себя: клиенту могла быть выдана команда для работы в режиме, который он не поддерживает, клиент мог принять пакет, содержащий некоторые параметры, которые вне диапазона или за пределами возможностей клиента, клиенту могла быть выдана команда перехода в режим в некорректной последовательности. Пакет отчета об ошибках клиента может использоваться для обнаружения ошибок во время нормальной работы, но наиболее полезным для разработчика и интегратора системы является диагностирование проблем при разработке и интеграции систем хоста и клиента. Клиент указывает свою возможность отправки Пакета отчета об ошибках клиента, используя значение параметра 142 в Списке ответа действительных параметров Пакета списка ответа действительных состояний.

Формат одного варианта осуществления Пакета отчета об ошибках клиента показан в основном на фиг.84А. Как показано на фиг.84А, Пакет отчета об ошибках клиента структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Количества элементов списка, Списка кодов ошибок и ЦИК. Значение Типа пакета 142 идентифицирует пакет в качестве Пакета отчета об ошибках клиента. Поле ИД cClient резервируется для будущего использования и, как правило, устанавливается в настоящее время в нуль. Поле (2 байта) Количества элементов списка задает количество элементов в следующем Списке кодов ошибок. Поле (в данном случае 8 байтов) Списка кодов ошибок представляет собой список, содержащий один или несколько элементов Списка отчета об ошибках. Формат одного элемента Списка отчета об ошибках показан на фиг.87В.

В одном варианте осуществления, как показано на фиг.87В, каждый Элемент списка отчета об ошибках имеет длину точно 4 байта и имеет структуру в одном варианте осуществления, содержащую: 2-байтовое поле Кода ошибки дисплея, которое задает тип сообщаемой ошибки, 2-байтовое поле Подкода ошибки задает большую степень детализации, касающуюся ошибки, определенной пакетом Кода ошибки клиента. Заданное определение каждого Кода ошибки клиента определяется изготовителем клиента. Подкод ошибки не должен определяться для каждого Кода ошибки дисплея, и в тех случаях, когда Подкод ошибки не определен, значение устанавливается в нуль. Заданное определение каждого Подкода ошибки определяется изготовителем клиента.

40. Пакет идентификации клиента

Пакет идентификации клиента дает возможность клиенту возвращать идентифицирующие данные в ответ на Пакет запроса заданного состояния. В одном варианте осуществления клиент указывает возможность отправки Пакета идентификации клиента, используя значение параметра 144 в Списке ответа действительных параметров Пакета списка ответа действительных состояний. Для хоста является полезным иметь возможность определения названия изготовителя устройства клиента и номер модели посредством считывания этих данных с клиента. Информация может использоваться для определения, имеет ли клиент специальные возможности, которые не могут быть описаны в Пакете возможностей клиента. Потенциально существует два способа, средства или механизма для считывания идентификационной информации с клиента. Один осуществляется посредством использования Пакета возможностей клиента, который содержит поля, аналогичные полям в базовой структуре расширенных данных идентификации дисплея (РДИД). Другой способ осуществляется посредством использования Пакета идентификации клиента, который содержит более богатый набор информации по сравнению с аналогичными полями в Пакете возможностей клиента. Это дает возможность хосту идентифицировать изготовителей, которым был назначен 3-символьный код расширенной архитектуры промышленного стандарта (РАПС), и позволяет серийным номерам содержать буквенно-цифровые символы.

Формат одного варианта осуществления Пакета идентификации клиента показан в основном на фиг.85. Как показано на фиг.85, Пакет идентификации клиента структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Недели изготовления, Года изготовления, Длины названия изготовителя, Длины названия продукта, Длины серийного номера, Строки названия изготовителя, Строки названия продукта, Строки серийного номера и ЦИК.

2-байтовое поле Типа пакета содержит значение, которое идентифицирует пакет в качестве Пакета идентификации клиента. Это значение выбирается равным 144 в одном варианте осуществления. Поле (2 байта) ИД cClient снова резервируется для будущего использования для ИД клиента и, как правило, устанавливается в нуль. Поле (2 байта) ЦИК содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

1-байтовое поле Недели изготовления содержит значение, которое определяет неделю изготовления дисплея. В по меньшей мере одном варианте осуществления это значение находится в диапазоне от 1 до 53, если оно поддерживается клиентом. Если это поле не поддерживается клиентом, тогда оно, как правило, устанавливается в нуль. 1-байтовое поле Года изготовления содержит значение, которое определяет год изготовления клиента (дисплея). Это значение представляет собой смещение от года 1990 в качестве начальной точки, хотя могут использоваться другие базовые годы. Годы в диапазоне от 1991 до 2245 могут быть выражены посредством этого поля. Пример: год 2003 соответствует значению Года изготовления 13. Если это поле не поддерживается клиентом, оно должно быть установлено на значение нуля.

Каждое из полей Длины названия изготовителя, Длины названия продукта и Длины серийного номера содержит 2-байтовые значения, которые задают длину поля Строки названия изготовителя, включая любые нулевые символы завершения или нулевые символы заполнения, длину поля Строки названия продукта, включая любые символы завершения или нулевые символы заполнения, и длину поля Строки серийного номера, включая любые нулевые символы завершения или нулевые символы заполнения, соответственно.

Каждое из полей Строки названия изготовителя, Строки названия продукта и Строки серийного номера содержит переменное количество байтов, задаваемое полями Длины названия изготовителя, названия продукта и серийного номера соответственно, которые содержат строку американского стандартного кода обмена информацией (АСКОИ), которая задает изготовителя, название продукта и буквенно-цифровой серийный номер дисплея соответственно. Каждая из этих строк завершается по меньшей мере одним нулевым символом.

41. Пакет возможностей альтернативного дисплея

Пакет возможностей альтернативного дисплея используется в качестве средства, структуры или способа для указания возможностей альтернативных дисплеев, присоединенных к контроллеру клиента ЦИМД. Он посылается в ответ на Пакет запроса заданного состояния. Когда устройство клиента выдает приглашение, оно посылает Пакет возможностей альтернативного дисплея для каждого альтернативного дисплея, который поддерживается. Если клиент имеет более одного альтернативного дисплея, тогда клиент должен посылать, генерировать или предоставлять многочисленные Пакеты возможностей альтернативного дисплея, один для каждого дисплея, в ответ на один Пакет запроса заданного состояния, хотя некоторые конфигурации могут использовать многочисленные Пакеты запроса заданного состояния, когда требуется, хотя это менее эффективно. Клиент может посылать Пакеты возможностей альтернативного дисплея в том, что может упоминаться как «непоследовательный порядок», основанный на значении поля Номера альтернативного дисплея. Клиент может указывать возможность отправки Пакета возможностей альтернативного дисплея при помощи значения параметра 145 в Списке ответа действительных параметров Пакета списка ответа действительных состояний.

Для ЦИМД-систем, работающих во внутреннем режиме, может быть общепринятым иметь более одного дисплея, подключенного к контроллеру клиента ЦИМД. Примерным

применением является мобильный телефон с большим дисплеем на внутренней стороне откидывающейся крышки и меньшим дисплеем на внешней стороне. Не является необходимым для клиента во внутреннем режиме возврат Пакета возможностей альтернативного дисплея по двум потенциальным причинам. Во-первых, хост может быть
 5 уже запрограммирован или иным образом проинформирован о возможностях во время изготовления, так как они используются в общем устройстве или корпусе. Во-вторых, вследствие их сборки клиент не может легко быть отсоединен или отделен от подключения к хосту, и хост может содержать жестко закодированную копию возможностей клиента, или по меньшей мере может иметь сведения, что они не меняются с изменением в клиенте, что
 10 могло иметь место иным образом.

Поле Количества альтернативных дисплеев Пакета возможностей клиента используется для сообщения, что присоединено более одного дисплея, и Пакет возможностей альтернативного дисплея сообщает возможности каждого альтернативного дисплея. Пакет видеопотока содержит 4 бита в поле Атрибутов данных пикселей для адресации каждого
 15 альтернативного дисплея в устройстве клиента.

Формат одного варианта осуществления Пакета возможностей альтернативного дисплея показан в основном на фиг.89. Как показано на фиг.86, Пакет возможностей альтернативного дисплея структурирован так, что имеет поля Длины пакета, Типа пакета, ИД cClient, Номера альтернативного дисплея, Резервированное 1, Ширины растрового
 20 изображения, Высоты растрового изображения, Ширины окна дисплея, Высоты окна дисплея, Разрядности карты цветов формата RGB, Возможностей формата RGB, Возможностей монохромного формата, Резервированное 2, Возможностей формата Y Cb Cr, Возможностей свойств дисплея, Резервированное 3 и ЦИК. Значение Типа пакета 145 идентифицирует пакет в качестве Пакета возможностей альтернативного дисплея. Поле ИД cClient резервируется для ИД клиента для будущего использования и, как правило,
 25 устанавливается в нуль.

Поле Количества альтернативных дисплеев использует 1 байт для указания идентификации альтернативного дисплея целым числом в диапазоне от 0 до 15. Первый альтернативный дисплей обычно обозначается как номер 0, и другие альтернативные
 30 дисплеи идентифицируются уникальными значениями Номера альтернативных дисплеев, причем наибольшее используемое значение представляет собой суммарное количество альтернативных дисплеев минус 1. Значения, большие чем суммарное количество альтернативных дисплеев минус 1, не используются. Пример: мобильный телефон, имеющий главный дисплей и дисплей ИД вызывающего абонента, подключенный к клиенту
 35 ЦИМД, имеет один альтернативный дисплей, так что Номер альтернативного дисплея у дисплея ИД вызывающего абонента, равен нулю, и поле Количества альтернативных дисплеев Пакета возможностей клиента имеет значение 1.

Поле (1 байт) Резервированное 1 резервируется для будущего использования. Все биты в этом поле устанавливаются в нуль. Одним назначением этого поля является то, чтобы
 40 вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

Поле Ширины растрового изображения использует 2 байта, которые задают ширину растрового изображения, выраженную в виде количества пикселей. Поле Высоты растрового изображения использует 2 байта, которые задают высоту растрового
 45 изображения, выраженную в виде количества пикселей. Поле Ширины окна дисплея использует 2 байта, которые задают ширину окна дисплея, выраженную в виде количества пикселей. Поле Высоты окна дисплея использует 2 байта, которые задают высоту окна дисплея, выраженную в виде количества пикселей.

Поле Разрядности карты цветов формата RGB использует 2 байта, которые задают
 50 количество битов красной, зеленой и синей цветовых составляющих, которые могут отображаться в режиме отображения карты цветов (палитры). Может использоваться максимум 8 битов для каждой цветовой составляющей (красной, зеленой и синей). Даже если 8 битов каждой цветовой составляющей посылаются в Пакете карты цветов,

используется только количество самых младших битов каждой цветовой составляющей, определенное в этом поле. Если клиент дисплея не может использовать формат карты цветов (палитры), тогда это значение равно нулю. Слово Разрядности карты цветов формата RGB состоит из трех отдельных значений без знака:

5 Биты 3-0 определяют максимальное количество битов синего цвета в каждом пикселе, причем значения 0-8 рассматриваются действительными. Биты 7-4 определяют максимальное количество битов зеленого цвета в каждом пикселе, причем значения 0-8 рассматриваются действительными. Биты 11-8 определяют максимальное количество битов красного цвета в каждом пикселе, причем значения 0-8 рассматриваются
10 действительными. Биты 14-12 резервируются для будущего использования и, как правило, устанавливаются в нуль. Бит 15 используется для указания возможности клиента принимать данные пикселей Карты цветов в упакованном или неупакованном формате. Когда Бит 15 установлен на уровень логической единицы, то это означает, что клиент может принимать данные пикселей Карты цветов или в упакованном, или в неупакованном
15 формате. Если бит 15 установлен в логический нуль, то это означает, что клиент может принимать данные пикселей Карты цветов только в неупакованном формате.

Поле возможностей формата RGB использует 2 байта для задания количества битов разрешения, которое может отображаться в формате RGB. В одном варианте осуществления, если клиент не может использовать формат RGB, тогда это значение
20 устанавливается равным нулю. Слово Возможностей формата RGB состоит из трех отдельных значений без знака: Биты 3-0 определяют максимальное количество битов синего цвета (яркость синего цвета) в каждом пикселе, Биты 7-4 определяют максимальное количество битов зеленого цвета (яркость зеленого цвета) в каждом пикселе, и Биты 11-8 определяют максимальное количество битов красного цвета (яркость
25 красного цвета) в каждом пикселе. Биты 14-12 резервируются для будущего использования и устанавливаются в нуль. Бит 15 используется для указания возможности клиента принимать RGB-данные пикселей в упакованном или неупакованном формате. Когда Бит 15 установлен на уровень логической единицы, то это означает, что клиент может принимать RGB-данные пикселей или в упакованном, или в неупакованном формате. Если
30 бит 15 установлен в логический нуль, то это означает, что клиент может принимать RGB-данные пикселей только в неупакованном формате.

1-байтовое поле Возможностей монохромного формата содержит значение или информацию для задания количества битов разрешения, которое может отображаться в монохромном формате. Если клиент не может использовать монохромный формат, тогда
35 это значение устанавливается равным нулю. Биты 6-4 резервируются для будущего использования и, как правило, устанавливаются в нуль. Биты 3-0 определяют максимальное количество битов серой шкалы, которая может существовать в каждом пикселе. Эти четыре бита делают возможным задавать то, что каждый пиксель состоит из 1 до 15 битов. Если значение равно нулю, тогда монохромный формат не поддерживается
40 клиентом. Бит 7, когда он установлен в единицу, указывает, что клиент может принимать данные пикселей монохромного формата или в упакованном, или в неупакованном формате. Если бит 7 установлен в нуль, то это означает, что клиент может принимать данные пикселей монохромного формата только в неупакованном формате.

Поле Резервированное 2 имеет поле длиной 1 байт, зарезервированное для будущего
45 использования и, как правило, имеет все биты в этом поле, установленные на уровень логического нуля. В одном варианте осуществления одним назначением этого поля является то, чтобы вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

50 2-байтовое поле Возможностей формата Y Cb Cr задает количество битов разрешения, которое может отображаться в формате Y Cb Cr. Если клиент не может использовать формат Y Cb Cr, тогда это значение равно нулю. Слово Возможностей формата Y Cb Cr состоит из трех отдельных значений без знака: Биты 3-0 определяют максимальное

количество битов, которые задают отсчет Cb, Биты 7-4 определяют максимальное количество битов, которые задают отсчет Cr, Биты 11-8 определяют максимальное количество битов, которые задают отсчет Y, и Биты 14-12 резервируются для будущего использования и устанавливаются в нуль. Бит 15, когда он установлен в единицу,

5 указывает, что клиент может принимать данные пикселей формата Y Cb Cr или в упакованном, или в неупакованном формате. Если бит 15 установлен в нуль, то это означает, что клиент может принимать данные пикселей формата Y Cb Cr только в неупакованном формате.

2-байтовое поле Возможностей байеровского формата задает количество битов разрешения, группу пикселей и порядок пикселей, которые могут пересылаться в байеровском формате. Если клиент не может использовать байеровский формат, тогда это значение устанавливается в нуль. Поле Возможностей байеровского формата состоит из следующих значений: Биты 3-0 определяют максимальное количество битов яркости, которая существует в каждом пикселе, Биты 5-4 определяют комбинацию группы пикселей, которая может потребоваться, Биты 8-6 определяют порядок пикселей, который требуется, и Биты 14-9 резервируются для будущего использования и устанавливаются в нуль. Бит 15, когда он установлен в единицу, указывает, что клиент может принимать данные пикселей байеровского формата или в упакованном, или в неупакованном формате. Если бит 15 установлен в нуль, это указывает, что клиент может принимать данные пикселей байеровского формата только в неупакованном формате.

2-байтовое поле ЦИК содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

42. Пакет доступа к регистрам

Пакет доступа к регистрам обеспечивает или хосту, или клиенту средство, механизм или способ для доступа к регистрам конфигурации и состояния на противоположном конце линии передачи данных ЦИМД. Регистры, вероятно, являются уникальными для каждого дисплея или контроллера устройства. Эти регистры уже существуют во многих дисплеях, которые требуют установки конфигураций, режимов работы и имеют другие полезные и необходимые установки. Пакет доступа к регистрам дает возможность хосту или клиенту ЦИМД как записывать в регистр, так и запрашивать считывание регистра, используя линию передачи данных ЦИМД. Когда хост или клиент запрашивает считывание регистра, противоположный конец должен ответить посылкой данных регистров в этом же типе пакета, но также указанием, что они представляют собой данные, считанные из конкретного регистра с использованием поля Считывания/записи информации. Пакет доступа к регистрам может использоваться для считывания или записи многочисленных регистров посредством задания числа регистров, большего 1. Клиент указывает возможность поддержки Пакета доступа к регистрам, используя бит 22 поля Возможностей свойств клиента Пакета возможностей клиента. Клиент использует пакет инкапсуляции для посылки Пакета доступа к регистрам, поэтому представляя то, что оказывается в качестве пакета в конфигурации или структуре пакета.

Формат одного варианта осуществления Пакета доступа к регистрам показан в основном на фиг.87. Как показано на фиг.87, Пакет доступа к регистрам структурирован так, что имеет поля Длины пакета, Типа пакета, ИД bClient, Флагов считывания/записи, Адреса регистра, ЦИК параметров, Списка данных регистров и ЦИК данных регистров. Значение 45 Типа пакета 146 идентифицирует пакет в качестве Пакета доступа к регистрам. Поле ИД bClient резервируется для будущего использования и в настоящее время, как правило, устанавливается в нуль.

2-байтовое поле Флагов считывания/записи задает заданный пакет в качестве или записи, или считывания, или ответа на считывание и предоставляет число значений 50 данных.

Биты 15-14 служат в качестве Флагов считывания/записи. Если Биты [15:14] равны «00», тогда этот пакет содержит данные, подлежащие записи в регистр, адресуемый полем Адреса регистра. Данные, подлежащие записи в заданные регистры, содержатся в поле

Списка данных регистров. Если Биты [15:14] равны «10», тогда это запрос данных из одного или нескольких регистров, адресуемых полем Адреса регистра. Если Биты [15:14] равны «11», тогда этот пакет содержит данные, которые были запрошены в ответ на Пакет доступа к регистрам, имеющий биты 15:14 Флагов считывания/записи, установленные в «10». Поле Адреса регистра содержит адрес регистра, соответствующего первому элементу Списка данных регистров, и поле Списка данных регистров содержит данные, которые были считаны по адресу или адресам. Если Биты [15:14] равны «01», то это рассматривается как недействительное значение, это значение резервируется для будущего использования и не используется в настоящее время, но для специалиста в данной области техники понятно, как использовать его для будущих применений.

Биты 13:0 используют 14-битовое целое число без знака для задания количества элементов 32-битовых Данных регистров, подлежащих пересылке в поле Списка данных регистров. Если биты 15:14 равны «00», тогда биты 13:0 задают количество элементов 32-битовых данных регистров, которые содержатся в поле Списка данных регистров, подлежащем записи в регистры, начиная с регистра, заданного полем Адреса регистра. Если биты 15:14 равны «10», тогда биты 13:0 задают количество элементов 32-битовых данных регистров, которые принимающее устройство посылает на устройство, запрашивающее, чтобы регистры были считаны. Поле Списка данных регистров в этом пакете не содержит элементов и имеет нулевую длину. Если биты 15:14 равны «11», тогда биты 13:0 задают количество элементов 32-битовых данных регистров, которые были считаны из регистров, которые содержатся в поле Списка данных регистров. Биты 15:14 в настоящее время не устанавливаются равными «01», что считается недействительным значением, и, иначе, резервируются для будущих обозначений или использования.

Поле Адреса регистра использует 4 байта для указания адреса регистра, по которому должен быть записан или считан. Для адресации регистров, адресация которых использует менее 32 бита, старшие биты устанавливаются в нуль.

2-байтовое поле ЦИК параметров содержит ЦИК всех байтов с Длины пакета до Адреса регистра. Если проверка этим ЦИК не успешна, тогда весь пакет отбрасывается.

Поле Списка данных регистров содержит список 4-байтовых значений данных регистров, подлежащих записи в регистры клиента, или значений, которые были считаны из регистров устройства клиента.

2-байтовое поле ЦИК данных регистров содержит ЦИК только Списка данных регистров. Если проверка этим ЦИК не успешна, тогда Данные регистров могут все же использоваться, но число ошибок ЦИК получает приращение.

D. ЦИК пакета

Поле ЦИК находится в конце пакетов и иногда после некоторых более критичных параметров в пакете, которые могут иметь значительно большее поле данных и, таким образом, повышенную вероятность ошибок по время пересылки. В пакетах, которые имеют два поля ЦИК, генератор ЦИК, когда используется только один, повторно инициализируется после первого ЦИК, так что на вычисления ЦИК, следующие за длинным полем данных, не оказывают влияние параметры в начале пакета.

Существует удаленная возможность получения хорошего ЦИК для пакетов, содержащих многочисленные ошибочные биты. Вероятность обнаружения хорошего ЦИК на пакетах с ошибками приближается к $7,6 \cdot 10^{-6}$ на очень длинных пакетах, содержащих много ошибок. Линия передачи данных ЦИМД преднамеренно имеет очень низкую или нулевую частоту появления ошибок. ЦИК предназначен для использования для контроля степени исправности линии передачи данных, и он не предназначен для обнаружения ошибок по заданным пакетам с целью определения, должны ли пакеты быть переданы повторно.

В примерном варианте осуществления полином, используемый для вычисления ЦИК, известен как ЦИК-16 или $X^{16}+X^{15}+X^2+X^0$. Примерная реализация генератора ЦИК и устройства 3600 проверки, полезных для реализации изобретения, показана на фиг.36. На фиг.36 регистр 3602 ЦИК инициализируется значением 0x0001 перед самой пересылкой первого бита пакета, который вводится на линию Tx_MDDI_Data_Before_CRC, затем байты

пакета сдвигаются в регистр, начиная сначала с СМБ. Отметьте, что номера битов регистра на этой фигуре соответствуют порядку используемого полинома, а не положениям битов, используемым ЦИМД. Более эффективным является сдвиг регистра ЦИК в одном направлении, и это приводит к тому, что бит 15 ЦИК появляется в положении 0 бита поля ЦИК ЦИМД, и бит 14 регистра ЦИК - в положении 1 бита поля ЦИК ЦИМД и т.д. до тех пор, пока не будет достигнуто положение 14 бита ЦИМД.

В качестве примера, если содержимым пакета для Пакетов запроса клиента и состояния являются: 0x000с, 0x0046, 0x000, 0x0400, 0x00, 0x00, 0x0000 (или представляется как последовательность байтов: 0x0с, 0x00, 0x46, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00) и представляются с использованием входных сигналов мультиплексоров 3604 и 3606 и вентиля 3608 И, то результирующим выходным сигналом ЦИК на линии Tx_MDDI_Data_With_CRC является 0xd9aa (или представляется как последовательность 0xaa, 0xd9).

Если генератор ЦИК и устройство 3600 проверки конфигурируется как устройство проверки ЦИК, то ЦИК, который принимается по линии Rx_MDDI_Data, вводится в мультиплексор 3604 и вентиль 3612 исключающее ИЛИ (XOR) и сравнивается побитно со значением, находящимся в регистре ЦИК, используя вентиль 3610 НЕ-ИЛИ, вентиль 3608 И и вентиль 3614 И. Если имеются какие-то ошибки в виде выходного сигнала вентиля 3614 И, ЦИК получает приращение один раз за каждый пакет, который содержит ошибку ЦИК посредством подключения выхода вентиля 3614 ко входу регистра 3602. Отметьте, что примерная схема, показанная на принципиальной схеме фиг.36, может выводить более одного сигнала ошибки ЦИК в течение данного окна CHECK_CRC_NOW (теперь проверить ЦИК) (см. фиг.37В). Поэтому счетчик ошибок ЦИК в основном подсчитывает только первую копию ошибки ЦИК в течение каждого интервала, где активен CHECK_CRC_NOW. Если он сконфигурирован как генератор ЦИК, то ЦИК тактируется из регистра ЦИК в момент времени, совпадающий с концом пакета.

Временные соотношения для входных и выходных сигналов и сигналов включения изображены графически на фиг.37А и 37В. Генерирование ЦИК и передача пакета данных показаны на фиг.37А с состоянием (0 или 1) сигналов Gen_Reset, Check_CRC_Now, Generate_CRC_Now (теперь генерировать ЦИК) и Sending_MDDI_Data (посылка данных ЦИМД) вместе с сигналами Tx_MDDI_Data_Before_CRC (передать данные ЦИМД перед ЦИК) и Tx_MDDI_Data_With_CRC (передать данные ЦИМД с ЦИК). Прием пакета данных и проверка значения ЦИК показаны на фиг.37D с состоянием сигналов Gen_Reset (сброс генератора), Check_CRC_Now (теперь проверить ЦИК), Generate_CRC_Now (теперь генерировать ЦИК) и Sending_MDDI_Data (посылка данных ЦИМД) вместе с сигналами Rx_MDDI_Data и ошибки ЦИК.

Е. Перегрузка кода ошибки для ЦИК пакета

Всякий раз когда только пакеты данных и ЦИК пересылаются между хостом и клиентом, нет предоставляемых кодов ошибок. Единственной ошибкой является потеря синхронизации. В противном случае приходится ожидать истечения времени ожидания линии передачи данных из-за отсутствия хорошего тракта пересылки данных или магистральной линии связи пересылки данных и затем возвращать линию передачи данных в исходное состояние и продолжать. К сожалению, это отнимает много времени и в некоторой степени неэффективно.

Для использования в одном варианте осуществления был разработан новый метод, при котором часть ЦИК пакетов используется для пересылки информации о кодах ошибок. Это в основном показано на фиг.65. Т.е. один или несколько кодов ошибок генерируются процессорами или устройствами, оперирующими пересылкой данных, которые указывают заданные предварительно определенные ошибки или дефекты, которые могут иметь место при обработке обмена данными или в линии передачи данных. Когда встречается ошибка, тогда соответствующий код ошибки генерируется и пересылается с использованием битов для ЦИК пакета. Т.е. значение ЦИК перегружается, или переписывается, требуемым кодом ошибки, который может быть обнаружен на приемном конце устройством контроля ошибок

или устройством проверки ошибок, которые контролируют значения поля ЦИК. Для этих случаев, при которых код ошибки совпадает со значением ЦИК по некоторой причине, дополнение ошибки пересылается для того, чтобы предотвратить путаницу.

В одном варианте осуществления, чтобы обеспечить систему надежного

5 предупреждения и обнаружения ошибок, код ошибки может пересылаться несколько раз, используя серию пакетов в основном все, которые пересылаются или посылаются после обнаружения ошибки. Это происходит до тех пор, пока из системы не будет устранено условие, создающее ошибку, в этот момент обыкновенные биты ЦИК пересылаются без перегрузки другим значением.

10 Этот метод перегрузки значения ЦИК обеспечивает более быструю реакцию на ошибки системы, в то же время используя минимальное количество дополнительных битов или полей.

Как показано на фиг.66, механизм или устройство 6600 переписывания ЦИК показано с использованием обнаружителя или средства 6602 обнаружения ошибок, которые могут

15 формировать часть другой схемы, ранее описанной или известной, обнаруживает присутствие или существование ошибок на линии или в процессе обмена данными. Генератор или средство 6604 кода ошибки, которое может быть сформировано как часть другой схемы или может использовать методы, такие как таблицы преобразования, для хранения предварительно выбранных сообщений об ошибке, генерирует один или

20 несколько кодов ошибок для указания заданных предварительно определенных ошибок или дефектов, которые были обнаружены как имеющие место. Легко понятно, что устройства 6602 и 6604 могут быть сформированы в виде единственной схемы или устройства, как требуется, или как часть запрограммированной последовательности этапов для других известных процессоров и элементов.

25 Блок сравнения или средство 6606 сравнения значения ЦИК показано для проверки, является ли выбранный код или коды ошибки одинаковыми со пересылаемым значением ЦИК. Если это так, тогда генератор, или средство, или устройство генерирования дополнения кода используется для предоставления дополнения кодов ошибок, чтобы их нельзя было спутать с исходной комбинацией или значением ЦИК и запутать или

30 усложнить схему обнаружения. Селектор, или средство, элемент или устройство 6610 выбора кода ошибки затем выбирает код или значение ошибки, которое оно желает вставить или переписать, или их соответствующие дополнения, что подходит. Устройство перезаписи или механизм или средство 6612 перезаписи ЦИК кода ошибки представляет собой устройство, которое принимает поток, пакеты данных и требуемые коды,

35 подлежащие вставке и перезаписи соответствующих или подходящих значений ЦИК, чтобы переслать требуемые коды ошибок на приемное устройство.

Как упомянуто, код ошибки может пересылаться несколько раз, используя серию пакетов, так что устройство 6612 перезаписи может использовать элементы хранения в

40 памяти, чтобы сохранять копии кодов во время обработки, или повторно вызывать эти коды из предыдущих элементов или других известных ячеек памяти, которые могут использоваться для запоминания или хранения их значений, когда необходимо или когда требуется.

Общая обработка, которую реализует механизм перезаписи по фиг.66, подробно показана на фиг.67А и 67В. На фиг.67А ошибка, одна или несколько, обнаруживается на

45 этапе 6702 в данных или процессе обмена данными, и код ошибки выбирается на этапе 6704 для указания этого состояния. Одновременно или в соответствующей точке подлежащее замене значение ЦИК проверяется на этапе 6706 и сравнивается с требуемым кодом ошибки на этапе 6708. Результат этого сравнения, как описано ранее, представляет собой определение, является ли или нет требуемый код или другие

50 характерные значения, одинаковым с текущим значением ЦИК. Если это так, тогда обработка переходит на этап 6712, где дополнение, или в некоторых случаях другое характерное значение, что требуется, выбирается в качестве кода для вставки. Если было определено, какие коды ошибок или значения должны быть вставлены на этапах 6710

и 6714, тогда соответствующий код выбирается для вставки. Эти этапы иллюстрируются как отдельные для целей ясности, но, как правило, представляют один вариант выбора, основанный на выходном результате принятия решения этапа 6708. И наконец, на этапе 6716 соответствующие значения перезаписываются в расположение ЦИК для пересылки с

5 пакетами, адресат которых определяется процессом.

На стороне приема пакета, как показано на фиг.67В, значения ЦИК пакета контролируются на этапе 6722. Как правило, значения ЦИК контролируются одним или несколькими процессами в системе для определения, имела ли место ошибка в пересылке данных и запрашивать ли или нет повторную передачу пакета или пакетов или запретить

10 ли дальнейшие операции и т.п., некоторые из которых описаны выше. Как часть такого контроля информация также может использоваться для сравнения значений с известными или предварительно выбранными кодами ошибок или характерными значениями и обнаружения присутствия ошибок. Альтернативно может быть реализован отдельный процесс обнаружения ошибок и устройство контроля. Если код оказывается

15 присутствующим, он извлекается или иным образом отмечается на этапе 6724 для дополнительной обработки. Определение может быть выполнено на этапе 6726, является ли он или нет фактическим кодом или дополнением, в этом случае используется дополнительный этап 6728 для перевода значения в требуемое значение кода. В любом случае результирующий извлеченный код, дополнение или другие восстановленные

20 значения затем используются для обнаружения того, какая ошибка произошла в переданном коде на этапе 6730.

V. Состояние «спячки» линии передачи данных

Линия передачи данных ЦИМД может быстро перейти в состояние «спячки» и быстро выполнить пробуждение из «спячки». Эта быстрота реагирования дает возможность

25 системе или устройству обмена данными часто переводить линию передачи данных ЦИМД в состояние «спячки» для снижения потребления мощности, так как она может очень быстро снова пробуждаться для использования. В одном варианте осуществления, когда клиент во внешнем режиме пробуждается из состояния «спячки» в первый раз, он делает это со скоростью передачи данных и с согласованием во времени строб-импульсов,

30 которое согласуется со скоростью передачи 1 Мбит/с, т.е. пара MDDI_Stb должна переключаться со скоростью 500 кГц. Если характеристики клиента были обнаружены или переданы хосту, тогда хост может пробудить линию передачи данных, как правило, с любой скоростью от 1 Мбит/с до максимальной скорости передачи, с которой клиент может работать. Клиенты во внутреннем режиме могут пробуждаться с любой скоростью

35 передачи, с которой могут работать как хост, так и клиент. Это также, как правило, применимо к первому разу, когда пробуждается клиент во внутреннем режиме.

В одном варианте осуществления, когда линия передачи данных пробуждается из состояния «спячки», хост и клиент обмениваются последовательностью импульсов. Эти импульсы могут быть обнаружены с использованием низкоскоростных линейных

40 приемников, которые потребляют только часть тока, который требуется для приема дифференциальными приемниками сигналов с максимальной рабочей скоростью линии передачи данных. Или хост, или клиент могут пробудить линию передачи данных, так что протокол пробуждения предназначен для обработки возможного состязания, которое может иметь место, если и хост, и клиент предпринимают одновременную попытку пробуждения.

45 Во время состояния «спячки» дифференциальные драйверы MDDI_Data и MDDI_Stb выключаются в высокоимпедансное состояние и дифференциальное напряжение на всех дифференциальных парах равно нулю вольт. Дифференциальные линейные приемники, используемые для обнаружения последовательности импульсов во время пробуждения из «спячки», имеют преднамеренное смещение напряжения. В одном варианте

50 осуществления порог между уровнем логической единицы и логическим нулем в этих приемниках равен примерно 125 мВ. Это вызывает нахождение невозбужденной дифференциальной пары на уровне логического нуля во время последовательности пробуждения линии передачи данных.

Чтобы перейти в Состояние «спячки», хост посылает 64 цикла MDDI_Stb после ЦИК Пакета закрытия линии передачи данных. Хост выключает выход MDDI_Data0 хоста в диапазоне 16-56 циклов MDDI_Stb (включая задержки на распространение выключения выхода) после ЦИК. Хост завершает посылку 64 циклов MDDI_Stb после ЦИК пакета

5 Закрытия линии передачи данных, перед тем как он иницирует последовательность пробуждения. В одном варианте осуществления иницируемое хостом пробуждение определяется как необходимость ожидания хостом по меньшей мере 100 нс после того, как MDDI_Data0 достигнет действительного уровня логической единицы перед возбуждающими импульсами на MDDI_Stb. В одном варианте осуществления клиент ожидает по меньшей

10 мере 60 циклов MDDI_Stb после ЦИК Пакета закрытия линии передачи данных, до того как он возбудит MDDI_Data0 уровнем логической единицы, чтобы предпринять попытку пробуждения хоста.

Чтобы «пробудить» из Состояния «спячки», выполняется несколько действий или процессов. Когда клиенту, в данном случае дисплею, необходимы данные или обмен

15 данными, обслуживание с хоста, он генерирует импульс запроса посредством возбуждения линии MDDI_Data0 в состояние логической единицы в течение примерно от 70 до 1000 мкс, в то же время MDDI_Stb является неактивной и сохраняет MDDI_Data0, возбужденной уровнем логической единицы в течение примерно 70 циклов MDDI_Stb (в диапазоне от 60 до 80) после чего MDDI_Stb становится активной, хотя могут использоваться другие

20 периоды времени, как потребуется. Клиент затем выключает драйвер MDDI_Data0 переводом его высокоимпедансное состояние.

Если MDDI_Stb является активной во время состояния «спячки», хотя маловероятно, тогда клиент может возбудить только MDDI_Data0 состоянием логической единицы в течение примерно 70 циклов (в диапазоне 60-80) MDDI_Stb. Это действие вызывает запуск

25 или перезапуск хостом трафика данных по прямой линии (208) передачи данных и опрос клиента в отношении его состояния.

Хост должен обнаружить присутствие импульса запроса и начинает последовательность запуска, сначала возбуждая MDDI_Stb уровнем логического нуля и MDDI_Data0 - логическим высоким уровнем в течение по меньшей мере примерно 200 нс. И затем при

30 переключении MDDI_Stb продолжает возбуждать MDDI_Data0 уровнем логической единицы в течение примерно 150 циклов MDDI_Stb (диапазон 140-160) и в логический нуль в течение примерно 50 циклов MDDI_Stb. Клиент не должен посылать импульс запроса на обслуживание, если он обнаруживает MDDI_Data0 в состоянии логической единицы в течение более 80 циклов MDDI_Stb. Когда клиент обнаруживает MDDI_Data0 на уровне

35 логической единицы в течение 60-80 циклов MDDI_Stb, он начинает поиск в течение интервала, когда хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 циклов MDDI_Stb. После того как хост возбуждает MDDI_Data0 уровнем логического нуля в течение продолжительности 50 циклов MDDI_Stb, тогда хост начинает посылку пакетов по линии передачи данных. Первым посланным пакетом является Пакет заголовка подкадра.

40 Клиент начинает поиск Пакета заголовка подкадра, после того как MDDI_Data0 будет находиться на уровне логического нуля в течение 40 циклов MDDI_Stb из интервала в 50 циклов. Ниже более подробно описывается сущность выбора периодов времени и допусков на временные интервалы, связанные с обработкой состояния «спячки» и последовательности запуска. (См. фиг.68A-C ниже.)

45 Хост может иницировать пробуждение первым включением MDDI_Stb и одновременно возбуждать ее уровнем логического нуля. MDDI_Stb не должна возбуждаться уровнем логической единицы до тех пор, пока не будут выводиться импульсы, как описано ниже. После того как MDDI_Stb достигает уровня логического нуля, хост включает MDDI_Data0 и одновременно возбуждает ее уровнем логической единицы. MDDI_Data0 не должна

50 возбуждаться уровнем логического нуля во время процесса пробуждения до интервала, когда она возбуждается уровнем логического нуля в течение интервала из 50 импульсов MDDI_Stb, как описано ниже. Хост должен ожидать по меньшей мере 200 нс, после того как MDDI_Data0 достигнет действительного уровня логической единицы перед

возбуждающими импульсами на MDDI_Stb. Эта временная зависимость имеет место при учете задержек на включение выхода наихудшего случая. Это, по существу, гарантирует, что клиент имеет достаточное время для полного включения своего приемника MDDI_Stb после пробуждения уровнем логической единицы на MDD_Data0, которая была возбуждена хостом.

Пример этапов обработки для типового события 3800 запроса на обслуживание без состязания изображен на фиг.38, где события обозначаются для удобства в иллюстрации, используя буквы A, B, C, D, E, F и G. Процесс начинается в точке A, когда хост посылает Пакет закрытия линии передачи данных на устройство клиента для информирования его, что линия передачи данных будет переходить в состояние «спячки» с низким потреблением мощности. На следующем этапе хост переходит в состояние «спячки» с низким потреблением мощности посредством выключения драйвера MDD_Data0 и установки драйвера MDDI_Stb в логический нуль, как показано в точке B. MDD_Data0 возбуждается уровнем логического нуля высокоимпедансной цепью смещения. Через некоторый период времени клиент посылает импульс запроса на обслуживание хосту посредством возбуждения MDDI_Data0 уровнем логической единицы, как видно в точке C. Хост все еще обеспечивает уровень логического нуля, используя высокоимпедансную цепь смещения, но драйвер в клиенте переводит линию на уровень логической единицы. В течение 50 мкс хост опознает импульс запроса на обслуживание и устанавливает уровень логической единицы на MDDI_Data0 посредством включения ее драйвера, что видно в точке D. Клиент затем прекращает попытки установления импульса запроса на обслуживание, и клиент переводит свой драйвер в высокоимпедансное состояние, что видно в точке E. Хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 мкс, как показано в точке F, и также начинает генерировать MDDI_Stb таким образом, который согласуется с уровнем логического нуля на MDDI_Data0. Клиент начинает поиск Пакета заголовка подкадра, после того как MDDI_Data0 будет находиться на уровне логического нуля в течение 40 циклов MDDI_Stb. После установки MDDI_Data0 на уровень логического нуля и возбуждения MDDI_Stb в течение 50 мкс хост начинает передавать данные по прямой линии передачи данных посредством посылки Пакета заголовка подкадра, как показано в точке G.

Аналогичный пример иллюстрируется на фиг.39, где запрос на обслуживание устанавливается после начала последовательности повторного запуска линии передачи данных, и события снова обозначаются с использованием букв A, B, C, D, E, F и G. На нем представлен сценарий наихудшего случая, когда импульс или сигнал запроса от клиента поступает ближе всего к тому моменту, когда происходит разрушение Пакета заголовка подкадра. Процесс начинается в точке A, когда хост снова посылает Пакет закрытия линии передачи данных на устройство клиента для информирования его о том, что линия передачи данных перейдет в состояние «спячки» с низким потреблением мощности. На следующем этапе хост переходит в состояние «спячки» с низким потреблением мощности посредством выключения драйвера MDDI_Data0 и установки драйвера MDDI_Stb на уровень логического нуля, как показано в точке B. Как и раньше, MDDI_Data0 возбуждается уровнем логического нуля посредством высокоимпедансной цепи смещения. Через некоторый период времени хост начинает последовательность повторного запуска линии передачи данных посредством возбуждения MDDI_Data0 уровнем логической единицы в течение 150 мкс, как показано в точке C. До истечения 50 мкс после начала последовательности повторного запуска линии передачи данных дисплей также устанавливает MDDI_Data0 в течение 70 мкс, как показано в точке D. Это происходит потому, что дисплей должен запросить обслуживание у хоста и он не опознает, что хост уже начал последовательность повторного запуска линии передачи данных. Клиент затем прекращает попытки установления импульса запроса на обслуживание, и клиент переводит свой драйвер в высокоимпедансное состояние, как показано в точке E. Хост продолжает возбуждать MDDI_Data0 уровнем логической единицы. Хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 мкс, как

показано в точке F, и также начинает генерировать MDDI_Stb таким образом, который согласуется с уровнем логического нуля на MDDI_Data0. После установки MDDI_Data0 на уровень логического нуля и возбуждения MDDI_Stb в течение 50 мкс хост начинает передавать данные по прямой линии передачи данных посредством посылки Пакета

5 заголовка подкадра, как показано в точке G.

Из вышеупомянутого описания видно, что известное решение включало в себя прохождение хостом двух состояний как часть последовательности пробуждения. В течение первого состояния хост возбуждает сигнал MDDI_Data0 высоким уровнем в течение 150 мкс и затем возбуждает сигнал MDDI_Data0 низким уровнем в течение 50 мкс, в то же время активизируя линию MDDI_Stb, и затем начинает передавать пакеты ЦИМД. Этот процесс хорошо работает, осуществляя прогресс в данной области техники, в смысле скоростей передачи данных, достигаемых с использованием устройства и способов ЦИМД. Однако, как указано ранее, большее быстроедействие на основе уменьшенного времени реакции на условия или способность более быстро выбирать следующий этап или процесс представляет собой возможность упрощения обработки или элементов и всегда пользуется спросом.

Заявители открыли новый подход, обладающий признаками изобретения, для обработки и согласования во времени пробуждения, в которых хост использует согласование во времени, основанное на тактовом цикле, для переключения сигнала. В этой конфигурации хост начинает переключение MDDI_Stb с 0 до 10 мкс, после того как хост возбуждает сигнал MDDI_Data0 высоким уровнем в начале последовательности пробуждения, и не ожидает до тех пор, пока сигнал будет переведен на низкий уровень. Во время последовательности пробуждения хост переключает MDDI_Stb, как если бы сигнал MDDI_Data0 всегда был на уровне логического нуля. Это эффективно снимает понятие времени со стороны клиента, и хост изменяет с известных периодов 150 мкс и 50 мкс для первых двух состояний на 150 тактовых циклов и 50 тактовых циклов для этих периодов.

Хост теперь становится ответственным за возбуждение этой линии данных высоким уровнем и в пределах 10 тактовых циклов за начало передачи строб-сигнала, как если бы линия данных имела нулевой уровень. После того как хост возбуждал линию данных высоким уровнем в течение 150 тактовых циклов, хост возбуждает линию данных низким уровнем в течение 50 тактовых циклов, в то же время продолжая передавать строб-сигнал. После того как он завершит оба этих процесса, хост может начинать передавать первый пакет заголовка подкадра.

На стороне клиента реализация клиента теперь может использовать генерируемые тактовые импульсы для вычисления количества тактовых циклов, в течение которых линия данных сначала имеет высокий уровень, а затем низкий. Количество тактовых циклов, которое должно произойти на линии данных, возбуждаемой состоянием с высоким уровнем, равно 150, и на линии данных, возбуждаемой низким уровнем, равно 50. Это означает, что для надлежащей последовательности пробуждения клиент должен иметь способность подсчета по меньшей мере 150 непрерывных тактовых циклов линии данных, равных высокому уровню, за которыми следуют по меньшей мере 50 непрерывных тактовых циклов линии данных, равных низкому уровню. Если эти два условия удовлетворяются, клиент может начинать поиск служебного слова первого подкадра. Прерывание этой комбинации используется как основа для возврата счетчиков в начальное состояние, в котором клиент снова выполняет поиск первых 150 непрерывных тактовых циклов линии данных, равных высокому уровню.

Реализация клиента изобретения для основанного хостом пробуждения из состояния «спячки» очень похожа на начальный случай запуска за исключением того, что тактовая частота не начинается принудительно с 1 Мбит/с, как описано ранее. Вместо этого тактовая частота может быть установлена для возобновления с любой предыдущей скорости, которая была активной, когда линия обмена данными перешла в состояние «спячки». Если хост начинает передачу строб-сигнала, как описано выше, клиент должен быть способен снова подсчитывать по меньшей мере 150 непрерывных тактовых циклов

линии данных, равных высокому уровню, за которыми следуют по меньшей мере 50 непрерывных тактовых циклов линии данных, равных низкому уровню. Если эти два условия были выполнены, клиент может начинать поиск служебного слова.

Реализация клиента изобретения для основанного клиентом пробуждения из состояния «спячки» аналогична основанному хостом пробуждению за исключением того, что оно начинается с того, что клиент возбуждает линию данных. Клиент может асинхронно возбуждать линию данных без тактовых импульсов для пробуждения устройства хоста. Если хост распознает, что линия данных возбуждается высоким уровнем от клиента, он может начинать свою последовательность пробуждения. Клиент может подсчитывать количество тактовых циклов, генерируемых хостом, начиная или в течение своего процесса пробуждения. Если клиент подсчитывает 70 непрерывных тактовых циклов данных, равных высокому уровню, он может остановить возбуждение линии данных высоким уровнем. В этот момент хост должен уже также возбуждать линию данных высоким уровнем. Клиент затем может подсчитывать другие 80 непрерывных тактовых циклов линии данных, равных высокому уровню, для достижения 150 тактовых циклов линии данных, равных высокому уровню, и затем может выполнять поиск 50 тактовых циклов линии данных, равных низкому уровню. Если эти три условия были выполнены, клиент может начинать поиск служебного слова.

Преимущество этой новой реализации обработки пробуждения заключается в том, что она устраняет потребность в устройстве измерения времени. Является ли оно генератором, или цепью разряда конденсатора, или другими такими известными устройствами, клиенту больше не нужны такие внешние устройства для определения условий запуска. Это экономит деньги и площадь схемы при реализации контроллеров, счетчиков и т.п. на плате устройства клиента. Хотя это может не быть таким выгодным для клиента, как для хоста, этот метод также должен потенциально упрощать хост в отношении логики со сверхвысокой плотностью (ЛСВП), используемой для схемы ядра. Потребление мощности использования линий данных и строб-линий в качестве уведомления пробуждения и источника измерения также будет меньше, так как не будет необходимости в работе внешних схем для элементов ядра для ожидания основанного хостом пробуждения. Количество используемых циклов или тактовых периодов является примерным и могут использоваться другие периоды, что будет очевидным для специалиста в данной области техники.

Преимущество этой новой реализации обработки пробуждения заключается в том, что она устраняет необходимость в устройстве измерения времени. Является ли оно генератором, или цепью разряда конденсатора, или другими такими известными устройствами, клиенту больше не нужны такие внешние устройства для определения условий запуска. Это экономит деньги и площадь схемы при реализации контроллеров, счетчиков и т.п. на плате устройства клиента. Хотя это может не быть таким выгодным для клиента, как для хоста, этот метод также должен потенциально упрощать хост в отношении логики со сверхвысокой плотностью (ЛСВП), используемой для схемы ядра. Потребление мощности использования линий данных и строб-линий в качестве уведомления пробуждения и источника измерения также будет меньше, так как не будет необходимости в работе внешних схем для элементов ядра для ожидания основанного хостом пробуждения.

Для разъяснения и иллюстрации принципа действия этого нового метода на фиг.68A, 68B и 68C показаны временные соотношения MDDI_Data0, MDDI_Stb и различные операции относительно тактовых циклов.

Пример этапов обработки для типового Иницируемого хостом пробуждения без состязания изображен на фиг.68A, где события снова для удобства иллюстрации обозначаются с использованием букв A, B, C, D, E, F и G. Процесс начинается в точке A, когда хост посылает Пакет закрытия линии передачи данных на устройство клиента для информирования его, что линия передачи данных переходит в состояние «спячки» с низким потреблением мощности. На следующем этапе, в точке B, хост переключает MDDI_Stb в

течение примерно 64 циклов (или сколько требуется для разработки системы), чтобы дать возможность обработке клиентом завершиться до прекращения переключения MDDI_Stb, которое останавливает восстановленные тактовые импульсы в устройстве клиента. Хост также первоначально устанавливает MDDI_Data0 на уровень логического нуля и затем

5 выключает выход MDDI_Data0 в диапазоне 16-48 циклов (включая в основном задержки на распространение выключения выхода) после ЦИК. Может быть желательным перевести быстродействующие приемники для MDDI_Data0 и MDDI_Stb в клиенте в состояние с

10 низким потреблением мощности через некоторое время после 48 циклов после ЦИК и до следующего этапа (С). Клиент переводит свои быстродействующие приемники для MDDI_Data0 и MDDI_Stb в состояние «спячки» в любой момент времени после переднего фронта 48-го цикла MDDI_Stb после ЦИК Пакета закрытия линии передачи данных. Рекомендуется, чтобы клиент переводил свои быстродействующие приемники для MDDI_Data0 и MDDI_Stb в состояние «спячки» до переднего фронта 64-го цикла MDDI_Stb после ЦИК Пакета закрытия линии передачи данных.

15 Хост переходит в состояние «спячки» с низким потреблением мощности в точке или на этапе С посредством выключения драйверов MDDI_Data0 и MDDI_Stb и перевода контроллера хоста в состояние «спячки» с низким потреблением мощности. Можно также установить драйвер MDDI_Stb на уровень логического нуля (используя высокоимпедансную цепь смещения) или продолжать переключение во время «спячки», как требуется. Клиент

20 также находится в состоянии «спячки» с низким уровнем потребления мощности.

После некоторого периода времени хост начинает последовательность повторного запуска линии передачи данных в точке D посредством включения выхода драйверов MDDI_Data0 и MDDI_Stb. Хост возбуждает MDDI_Data0 уровнем логической единицы и MDDI_Stb уровнем логического нуля до тех пор, пока драйверы не включают полностью свои

25 соответствующие выходы. Хост обычно ожидает примерно 200 наносекунд, после того как эти выходы достигнут требуемых логических уровней, перед возбуждением импульсов на MDDI_Stb. Это предоставляет клиенту время для подготовки к приему.

С включенными драйверами хоста и MDDI_Data0, возбужденной уровнем логической единицы, хост начинает переключать MDDI_Stb в продолжении 150 циклов MDDI_Stb, что

30 видно в точке E. Хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 циклов, как показано в точке F, и клиент начинает поиск Пакета заголовка подкадра, после того как MDDI_Data0 будет находиться на уровне логического нуля в течение 40 циклов MDDI_Stb. Хост начинает передавать данные по прямой линии передачи данных посредством посылки Пакета заголовка подкадра, как показано в точке G.

35 Пример этапов обработки для типового Иницируемого клиентом пробуждения без состязания изображен на фиг.68В, где события снова для удобства иллюстрации обозначаются с использованием букв A, B, C, D, E, F, G, H и I. Как и раньше, процесс начинается в точке A, когда хост посылает Пакет закрытия линии передачи данных для информирования клиента, что линия передачи данных переходит в состояние с низким

40 потреблением мощности.

В точке B хост переключает MDDI_Stb в течение примерно 64 циклов (или сколько требуется для разработки системы), чтобы дать возможность обработке клиентом завершиться до прекращения переключения MDDI_Stb, которое останавливает восстановленные тактовые импульсы в устройстве клиента. Хост также первоначально

45 устанавливает MDDI_Data0 на уровень логического нуля и затем выключает выход MDDI_Data0 в диапазоне 16-48 циклов (включая в основном задержки на распространение выключения выхода) после ЦИК. Может быть желательным перевести быстродействующие приемники для MDDI_Data0 и MDDI_Stb в клиенте в состояние с низким потреблением мощности через некоторое время после 48 циклов после ЦИК и до следующего этапа (С).

50 Хост переходит в состояние «спячки» с низким потреблением мощности в точке или на этапе С посредством выключения драйверов MDDI_Data0 и MDDI_Stb и перевода контроллера хоста в состояние «спячки» с низким потреблением мощности. Можно также установить драйвер MDDI_Stb на уровень логического нуля (используя высокоимпедансную

цель смещения) или продолжать переключение во время состояния «спячки», как требуется. Клиент также находится в состоянии «спячки» с низким уровнем потребления мощности.

После некоторого периода времени клиент начинает последовательность повторного запуска линии передачи данных в точке D посредством включения приемника MDDI_Stb и также включения смещения в приемнике MDDI_Stb для гарантирования того, что состоянием принятого варианта MDDI_Stb является уровень логического нуля в клиенте, перед тем как хост включит свой драйвер MDDI_Stb. Может быть желательным, чтобы клиент включил смещение несколько раньше включения приемника для гарантирования приема действительного дифференциального сигнала и подавления ошибочных сигналов, как требуется. Клиент включает драйвер MDDI_Data0, в то же время возбуждая линию MDDI_Data0 уровнем логической единицы. Допускается одновременное включение MDDI_Data0 и MDDI_Stb, если время для включения смещения и включения стандартного дифференциального приемника MDDI_Stb равно менее 200 нс.

В течение примерно 1 мс, точка E, хост распознает импульс запроса на обслуживание от клиента, и хост начинает последовательность повторного запуска линии передачи данных посредством включения выходов драйверов MDDI_Data0 и MDDI_Stb. Хост возбуждает MDDI_Data0 уровнем логической единицы и MDDI_Stb уровнем логического нуля в течение времени, необходимого для того, чтобы драйверы включили свои соответствующие выходы. Хост обычно ожидает примерно 200 наносекунд, после того как эти выходы достигнут требуемых логических уровней, перед возбуждением импульсов на MDDI_Stb. Это предоставляет клиенту время для подготовки к приему.

С включенными драйверами хоста и MDDI_Data0, возбужденной уровнем логической единицы, хост начинает выводить импульсы на MDDI_Stb в продолжении 150 циклов MDDI_Stb, что видно в точке F. Когда клиент распознает первый импульс на MDDI_Stb, он выключает смещение в своем приемнике MDDI_Stb. Клиент продолжает возбуждать MDDI_Data0 уровнем логической единицы в течение 70 циклов MDDI_Stb и выключает свой драйвер MDDI_Data0 в точке G. Хост продолжает возбуждать MDDI_Data0 уровнем логической единицы в продолжении дополнительных 80 импульсов MDDI_Stb и в точке H возбуждает MDDI_Data0 уровнем логического нуля.

Как видно в точках G и H, хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 циклов, и клиент начинает поиск Пакета заголовка подкадра, после того как MDDI_Data0 будет находиться на уровне логического нуля в течение 40 циклов MDDI_Stb. После возбуждения MDDI_Stb в течение длительности 50 циклов хост начинает передавать данные по прямой линии передачи данных посредством посылки Пакета заголовка подкадра, как показано в точке I.

Пример этапов обработки для типового Иницируемого хостом пробуждения с состоянием от клиента, т.е. клиент тоже желает пробудить линию передачи данных, изображен на фиг.68C. События снова для удобства иллюстрации обозначаются с использованием букв A, B, C, D, E, F, G, H и I. Как и раньше, процесс начинается в точке A, когда хост посылает Пакет закрытия линии передачи данных для информирования клиента, что линия передачи данных переходит в состояние с низким потреблением мощности, переходит к точке B, где MDDI_Stb переключается в течение примерно 64 циклов (или сколько требуется для разработки системы), чтобы дать возможность обработке клиентом завершиться, и затем в точку C, где хост переходит в состояние «спячки» с низким потреблением мощности посредством выключения драйверов MDDI_Data0 и MDDI_Stb и перевода контроллера хоста в состояние «спячки» с низким потреблением мощности. После некоторого периода времени хост начинает последовательность повторного запуска линии передачи данных в точке D посредством включения выхода драйвера MDDI_Data0 и MDDI_Stb и начинает переключать MDDI_Stb в течение длительности 150 циклов MDDI_Stb, что видно в точке E.

До 70 циклов MDDI_Stb после точки E, в данном случае точка F, клиент еще не распознал, что хост возбуждает MDDI_Data0 уровнем логической единицы, поэтому клиент

также возбуждает MDDI_Data0 уровнем логической единицы. Это происходит здесь, так как клиент имеет желание запросить обслуживание, но не распознает, что хост, с которым он пытается выполнить обмен данными, уже начал последовательность повторного запуска линии передачи данных. В точке G клиент прекращает возбуждать MDDI_Data0 и

5 переводит ее драйвер в высокоимпедансное состояние посредством выключения своего выхода. Хост продолжает возбуждать MDDI_Data0 уровнем логической единицы в течение 80 дополнительных циклов.

Хост возбуждает MDDI_Data0 уровнем логического нуля в течение 50 циклов, как показано в точке H, и клиент начинает поиск Пакета заголовка подкадра, после того как
10 MDDI_Data0 будет находиться на уровне логического нуля в течение 40 циклов MDDI_Stb. Хост начинает передавать данные по прямой линии передачи данных посредством посылки Пакета заголовка подкадра, как показано в точке I.

VI. Электрические спецификации интерфейса

В примерных вариантах осуществления Данные в формате Без возврата к нулю (BВН)
15 кодируются с использованием сигнала строба данных или формата DATA-STB, который позволяет встраивать тактовую информацию в сигналы данных и строба. Тактовые импульсы могут восстанавливаться без сложной системы фазовой автоподстройки частоты. Данные переносятся по двунаправленной дифференциальной линии передачи данных, обычно реализуемой с использованием кабеля проводной линии, хотя, как указано
20 ранее, могут использоваться другие проводники, печатные проводники или элементы пересылки. Строб-сигнал (STB) передается по однонаправленной линии передачи данных, которая возбуждается только хостом. Строб-сигнал переключает значение (0 или 1), всякий раз когда присутствует последовательное состояние, 0 или 1, которое остается одинаковым на линии или сигнале Data.

На фиг.40 в графическом виде показан пример того, как последовательность данных, такая как биты «1110001011», может быть передана с использованием кодирования DATA-STB. На фиг.40 сигнал 4002 DATA показан на верхней строке временной диаграммы сигналов, и сигнал 4004 STB показан на второй строке, при этом каждый момент времени выровнен соответствующим образом (общая точка начала). По мере прохождения
30 времени, когда происходит смена состояния, имеющая место на линии 4002 (сигнале) DATA, тогда линия 4004 (сигнал) STB сохраняет предыдущее состояние, таким образом, первое состояние «1» сигнала DATA сопоставляется с первым состоянием «0» сигнала STB, его начальным значением. Однако если или когда состояние, уровень сигнала DATA не меняется, тогда сигнал STB переключается в противоположное состояние, или «1» в
35 настоящем примере, как в случае на фиг.40, где DATA обеспечивают другое значение «1». Т.е. существует один и только один переход в течение цикла бита между DATA и STB. Поэтому сигнал STB переходит снова, на этот раз в «0», когда сигнал DATA остается в «1» и сохраняет этот уровень или значение, когда сигнал DATA изменяет уровень в «0». Когда сигнал DATA остается в «1», сигнал STB переключается в противоположное
40 состояние, или в «1» в настоящем примере, и т.д., когда сигнал DATA изменяет или сохраняет уровни или значения.

При приеме этих сигналов выполняется операция исключающее ИЛИ (XOR) над сигналами DATA и STB и получается тактовый сигнал 4006, который показан в нижней части временной диаграммы для относительного сравнения с требуемыми сигналами
45 данных и строба. На фиг.41 показан пример схемы, полезной для генерирования выходов или сигналов DATA и STB из входных данных хоста, а затем восстановления или вторичного получения данных из сигналов DATA и STB на клиенте.

На фиг.41 передающая часть 4100 используется для генерирования и передачи исходных сигналов DATA и STB по промежуточному тракту 4102 сигнала, тогда как
50 приемная часть 4120 используется для приема сигналов и восстановления данных. Как показано на фиг.41, для того чтобы пересылать данные с хоста на клиент, сигнал DATA подается на вход двух схемных элементов 4104 и 4106 триггеров D-типа вместе с тактовым сигналом для запуска схем. Два выходных сигнала (Q) триггерных схем затем

разделяются на дифференциальную пару сигналов MDDI_Data0+, MDDI_Data0- и MDDI_Stb+, MDDI_Stb- соответственно, используя два дифференциальных линейных драйвера 4108 и 4110 (режим напряжения). Вентиль, схема или логический элемент 4112

исключающего НЕ-ИЛИ (XNOR) с тремя входами подсоединен для приема Data и выходных

5 сигналов обоих триггеров и генерирует выходной сигнал, который обеспечивает ввод данных для второго триггера, который, в свою очередь, генерирует сигналы MDDI_Stb+, MDDI_Stb-. Для удобства вентиль исключаящее НЕ-ИЛИ имеет кружок инвертирования, расположенный для индикации того, что он действительно инвертирует выходной сигнал Q триггера, который генерирует строб-сигнал.

10 В приемной части 4120 на фиг.41 сигналы MDDI_Data0+, MDDI_Data0- и MDDI_Stb+, MDDI_Stb- принимаются каждым из двух дифференциальных линейных приемников 4122 и 4124, которые генерируют одиночные выходные сигналы из дифференциальных сигналов. Выходные сигналы усилителей затем подаются на каждый вход вентиля, схемы или логического элемента 4126

15 исключаящего ИЛИ с двумя входами, который вырабатывает тактовый сигнал. Тактовый сигнал используется для запуска каждой из двух схем 4128 и 4130 триггеров D-типа, которые принимают задержанную копию сигнала DATA через элемент 4132 задержки, при этом один из них (4128) генерирует значения «0» данных и другой (4130) - значения «1» данных. Тактовый сигнал также имеет независимый выход с логического элемента исключаящее ИЛИ. Так как тактовая информация распределяется

20 между линиями DATA и STB, никакой из переходов сигнала между состояниями не имеет большую частоту, чем половина частоты тактовых импульсов. Так как тактовые импульсы воспроизводятся с использованием операции исключаящее ИЛИ сигналов DATA и STB, система эффективно выдерживает двойную величину перекося между входными данными и тактовыми импульсами по сравнению со случаем, когда тактовый сигнал посылается

25 непосредственно по одинарной выделенной линии данных.

Сигналы пар MDDI_Data, MDDI_Stb+ и MDDI_Stb- работают в дифференциальном режиме, чтобы максимизировать защищенность от отрицательного воздействия шума. Каждая дифференциальная пара завершается параллельной согласованной нагрузкой с

30 характеристическим сопротивлением кабеля или проводника, используемого для пересылки сигналов. Как правило, все параллельные согласованные нагрузки находятся в устройстве клиента. Она находится около дифференциального приемника для прямого трафика (данных, посылаемых от хоста клиенту), но она находится на возбуждающем конце кабеля, или других проводников, или элементов пересылки для обратного трафика (данных, посылаемых от клиента хосту). Для обратного трафика сигнал возбуждается

35 клиентом, отражается высокоимпедансным приемником на хосте и завершается согласованной нагрузкой на клиенте. Как описано в другом месте, обратные данные или данные по обратной линии передачи данных могут пересылаться или посылаться со скоростями передачи данных, большей чем обратная величина задержки на двойное прохождение в кабеле. Проводники или сигналы MDDI_Stb+ и MDDI_Stb- возбуждаются

40 только хостом.

На фиг.42 показана примерная конфигурация элементов, используемых для выполнения драйверов, приемников и согласованных нагрузок для пересылки сигналов как части

интерфейса ЦИМД изобретения. Этот примерный интерфейс использует считывание

45 низкого напряжения, в данном случае 200 мВ, с перепадами мощности менее 1 вольта и малым потреблением мощности. Драйвер каждой сигнальной пары имеет выход дифференциального тока. При приеме пакетов ЦИМД пары MDDI_Data и MDDI_Stb используют обычный дифференциальный приемник с порогом дифференциального напряжения в нуль вольт. В состоянии «спячки» выходы драйвера выключаются, и резисторы параллельной согласованной нагрузки понижают дифференциальное

50 напряжение на каждой сигнальной паре до нуля вольт. Во время «спячки» специальный приемник на паре MDDI_Data0 имеет порог входного дифференциального напряжения смещения, равный положительным 125 мВ, который вызывает интерпретирование линейным приемником в состоянии «спячки» невозбужденной сигнальной пары в качестве

уровня логического нуля.

Дифференциальное напряжение дифференциальной пары определяется как разность напряжения положительного (+) сигнала минус напряжение отрицательного (-) сигнала. Названия сигналов дифференциальной пары оканчивается или «+», или «-», который

указывает положительный или отрицательный сигнал пары соответственно. Выходной ток драйвера дифференциальной пары определяется как ток, вытекающий из положительного (+) выхода. Ток, проходящий через отрицательный (-) выход дифференциального драйвера всегда равен по величине, но противоположен по направлению по сравнению с током, проходящим через положительный (+) выход этого же дифференциального драйвера.

Иногда хост или клиент одновременно возбуждают дифференциальную пару уровнем логической единицы или уровнем логического нуля, чтобы гарантировать действительный логический уровень на паре, когда направление потока данных изменяется (от хоста к клиенту или от клиента к хосту). Диапазон выходного напряжения и выходные технические данные все же выполняются с одновременно возбуждаемыми выходами, возбуждаемыми до одинакового логического уровня. В некоторых системах может быть необходимым возбуждать небольшой ток в согласованной дифференциальной паре для создания небольшого напряжения смещения в некоторые моменты времени во время «спячки» и когда линия передачи данных пробуждается из состояния «спячки». В некоторых ситуациях включенные цепи смещения тока смещения возбуждают уровни тока, упоминаемые как:

$I_{\text{ESD-and-Rx}}$ - внутренний диод для защиты от электростатического разряда (ЭСР) и вход дифференциального приемника, причем обычно $I_{\text{ESD-and-Rx}} \leq 1$ мкА;

$I_{\text{Tx-Hi-Z}}$ - выход дифференциального драйвера в высокоимпедансном состоянии, причем обычно $I_{\text{Tx-Hi-Z}} \leq 1$ мкА; и $I_{\text{external-ESD}}$ - утечка через внешние диоды защиты от ЭСР, причем обычно $I_{\text{external-ESD}} \leq 3$ мкА.

Каждый из этих токов утечки изображен на фиг.47. Цепи повышения напряжения и понижения напряжения должны достигать минимального дифференциального напряжения при условиях утечки наихудшего случая, описанных выше, когда все происходят одновременно. Общая утечка составляет ≤ 4 мкА для внутреннего режима без внешних диодов защиты от ЭСР и ≤ 10 мкА для внешнего режима с внешней защитой от ЭСР.

Электрические параметры и характеристики дифференциальных линейных драйверов и линейных приемников описываются для одного примерного варианта осуществления в таблицах VIIa-VIId. В функциональном отношении драйвер пересылает логический уровень на входе непосредственно на положительный выход, и обратный логический уровень входа - на отрицательный выход. Задержка от входа до выходов хорошо согласована с дифференциальной линией, которая возбуждается дифференциально. В большинстве реализациях перепад напряжения на выходах меньше перепада на входе для минимизации потребляемой мощности и электромагнитных излучений. В одном варианте осуществления минимальный перепад напряжения составляет примерно 0,5 В. Однако могут использоваться другие значения, что должно быть известно специалистам в данной области техники, и в некоторых вариантах осуществления изобретатели предполагают меньшее значение в зависимости от конструктивных ограничений.

Дифференциальные линейные приемники имеют те же характеристики, что и высокоскоростной компаратор напряжения. На фиг.41 входом без кружочка является положительный вход и входом с кружочком является отрицательный вход. На выходе присутствует логическая единица, если: $(V_{\text{input}}+) - (V_{\text{input}}-)$ больше нуля. Другой путь описания - это дифференциальный усилитель с очень большим (фактически бесконечным) коэффициентом усиления, причем выходной сигнал ограничивается на уровнях напряжения логического 0 и 1.

Перекус задержки между дифференциальными парами должен минимизироваться для работы дифференциальной системы передачи с наивысшей потенциальной скоростью.

Таблица VIIa Электрические спецификации передатчика хоста				
Параметр	Описание	Мин.	Макс.	Единицы

5	V _{output-Range}	Допустимый диапазон выходного напряжения драйвера хоста относительно заземления хоста	0,35	1,60	В
	I _{OD+}	Ток высокого уровня дифференциального выхода драйвера, соответствующий уровню логической единицы (при возбуждении согласованной линии передачи)	2,5	4,5	мА
	I _{OD-}	Ток низкого уровня дифференциального выхода драйвера, соответствующий уровню логического нуля (при возбуждении согласованной линии передачи)	-4,5	-2,5	мА
	T _{Rise-Fall}	Время нарастания и спада (между 20% и 80% амплитудами) выхода драйвера, измеренное в дифференциальном режиме	425	Прим.1	нс
	T _{skew-pair}	Перекус между положительным и отрицательным выходами одной и той же дифференциальной пары (перекус внутри пары)		125	нс
10	T _{Differential-Skew}	Пиковый перекус задержки между одной дифференциальной парой и любой другой дифференциальной парой		См. выше	нс
	T _A	Дрожание, граница бита относительно пересечения центра	0	T _B -283	нс
	T _{B-TP0-DRVR}	Дрожание, граница бита относительно минимального выходного уровня	0	См. выше	нс

Примечание 1: Максимальное время нарастания и спада составляет или 30% интервала передачи одного бита по одной дифференциальной паре, или 100 нс, что меньше.

		Таблица VIb Электрические спецификации передатчика клиента			
15	Параметр	Описание	Мин.	Макс.	Единицы
20	V _{output-Range-Ext}	Допустимый диапазон выходного напряжения драйвера клиента относительно заземления клиента (Внешний режим)	0	1,25	В
	V _{output-Range-Int}	Допустимый диапазон выходного напряжения драйвера клиента относительно заземления клиента (Внутренний режим)	0,35	1,60	В
	I _{OD+}	Ток высокого уровня дифференциального выхода драйвера, соответствующий уровню логической единицы (при возбуждении эквивалента цепей повышения напряжения и понижения напряжения, которые существуют на хосте и клиенте)	2,5	4,5	мА
25	I _{OD-}	Ток низкого уровня дифференциального выхода драйвера, соответствующий уровню логического нуля (при возбуждении эквивалента цепей повышения напряжения и понижения напряжения, которые существуют на хосте и клиенте)	-4,5	-2,5	мА
	T _{Rise-Fall}	Время нарастания и спада (между 20% и 80% амплитудами) выхода драйвера, измеренное в дифференциальном режиме	425	Прим.1	нс
	T _{skew-pair}	Перекус между положительным и отрицательным выходами одной и той же дифференциальной пары (перекус внутри пары)		125	нс
30	T _{Differential-Skew}	Пиковый перекус задержки между одной дифференциальной парой и любой другой дифференциальной парой		См. выше	нс
	T _A	Дрожание, граница бита относительно пересечения центра		T _B -283	нс
	T _{B-TP4-DRVR}	Дрожание, граница бита относительно минимального выходного уровня		См. выше	нс
Примечание 1: Максимальное время нарастания и спада составляет 30% интервала передачи одного бита по одной дифференциальной паре или 100 нс, что меньше.					

		Таблица VIc Электрические спецификации приемника клиента			
35	Параметр	Описание	Мин.	Тип.	Макс. Единицы
40	V _{IT+}	Пороговое напряжение высокого уровня дифференциального входа приемника. Выше этого дифференциального напряжения входной сигнал интерпретируется как уровень логической единицы.		0	50 мВ
	V _{IT-}	Пороговое напряжение низкого уровня дифференциального входа приемника. Ниже этого дифференциального напряжения входной сигнал интерпретируется как уровень логического нуля.	-50	0	мВ
	V _{IT+}	Пороговое напряжение высокого уровня дифференциального входа приемника (смещенное для пробуждения из «спячки»). Выше этого дифференциального напряжения входной сигнал интерпретируется как уровень логической единицы.		125	175 мВ
45	V _{IT-}	Пороговое напряжение низкого уровня дифференциального входа приемника (смещенное для пробуждения из «спячки»). Ниже этого дифференциального напряжения входной сигнал интерпретируется как уровень логического нуля.	75	125	мВ
	V _{Input-Range}	Допустимый диапазон входного напряжения приемника относительно заземления клиента.	0		1,65 В
	R _{term}	Значение сопротивления параллельной согласованной нагрузки	98	100	102 Ом
50	I _{in}	Входной ток утечки	-10		10 мкА
	C _{pad}	Емкость контактной площадки относительно заземления клиента (Прим. 1)			5 пФ
	C _{diff}	Емкость между двумя сигналами дифференциальной пары (Прим. 1)			1 пФ
	T _{skew-pair-INT}	Перекус, вызванный дифференциальным приемником между положительным и отрицательным входами дифференциального приемника этой же дифференциальной пары (перекус внутри пары). Внутренний режим			250 нс
	T _{skew-pair-EXT}	Перекус внутри пары, Внешний режим			50 нс

T Differential- Skew	Пиковый перекося задержки между одной дифференциальной парой и любой другой дифференциальной парой			См. выше	пс
T _A	Дрожание, граница бита относительно пересечения центра			T _B -38,5	пс
T _B -TP4- RCVR-INT	Дрожание, граница бита относительно минимального входного уровня (Внутренний режим)	0		См. выше	пс
T _B -TP4- RCVR-EXT	Дрожание, граница бита относительно минимального входного уровня (Внешний режим)	0		См. выше	пс

Таблица VIId Электрические спецификации приемника хоста					
Параметр	Описание	Мин.	Тип.	Макс.	Единицы
V _{IT+}	Пороговое напряжение высокого уровня дифференциального входа приемника (несмещенное). Выше этого дифференциального напряжения входной сигнал интерпретируется как уровень логической единицы.		0	50	мВ
V _{IT-}	Пороговое напряжение низкого уровня дифференциального входа приемника (несмещенное). Ниже этого дифференциального напряжения входной сигнал интерпретируется как уровень логического нуля.	-50	0		мВ
V _{IT+}	Пороговое напряжение высокого уровня дифференциального входа приемника (смещенное для пробуждения из «спячки»). Выше этого дифференциального напряжения входной сигнал интерпретируется как уровень логической единицы.		125	175	мВ
V _{IT-}	Пороговое напряжение низкого уровня дифференциального входа приемника (смещенное для пробуждения из «спячки»). Ниже этого дифференциального напряжения входной сигнал интерпретируется как уровень логического нуля.	75	125		мВ
V _{Input-Range}	Допустимый диапазон входного напряжения приемника относительно заземления хоста.	0		1,65	В
I _{In}	Входной ток утечки (исключая смещения при «спячке»)	-10		10	мкА
C _{pad}	Емкость контактной площадки относительно заземления хоста			5	пФ
C _{diff}	Емкость между двумя сигналами дифференциальной пары			1	пФ
T _{skew-pair}	Перекося, вызванный дифференциальным приемником между положительным и отрицательным входами дифференциального приемника одной и той же дифференциальной пары (перекося внутри пары).			250	пс
T _{skew-pair-EXT}	Перекося внутри пары, Внешний режим			50	пс
T _A	Дрожание, граница бита относительно пересечения центра			T _B -38,5	пс
T _B -TP0- RCVR-INT	Дрожание, граница бита относительно минимального выходного уровня (Внешний режим)			См. выше	пс
T _B -TP0- RCVR-EXT	Дрожание, граница бита относительно минимального выходного уровня (Внешний режим)			См. выше	пс

На фиг.42 контроллер 4202 хоста и контроллер 4204 клиента или дисплея показаны пересылающими пакеты по линии 4206 обмена данными. Контроллер хоста использует группу из трех драйверов 4210, 4212 и 4214 для приема сигналов DATA и STB хоста, подлежащих пересылке, а также для приема сигналов DATA клиента, подлежащих пересылке, в то время как клиент использует три драйвера 4230, 4232 и 4234. Драйвер, ответственный за прохождение сигнала DATA (4212) хоста, использует вход сигнала включения для активизации линии обмена данными в основном только тогда, когда требуется пересылка с хоста на клиент. Так как сигнал STB формируется как часть пересылки данных, то для этого драйвера (4212) не используется дополнительный сигнал включения. Входы каждого приемника (4132, 4230) DATA и STB клиента имеют импедансы или резисторы 4218 и 4220 согласованной нагрузки, соответственно установленные параллельно им. Драйвер 4234 в контроллере клиента используется для подготовки сигналов данных, пересылаемых с клиента на хост, где драйвер 4214 на входной стороне обрабатывает данные.

Специальные приемники (драйвера) 4216 и 4236 подсоединены или подключены к линиям DATA и генерирует или используют смещение напряжения 125 мВ, ранее описанное как часть управления состоянием «спячки», описанного в другом месте. Смещения вызывают интерпретирование линейными приемниками в состоянии «спячки» невозбужденных сигнальных пар в качестве уровня логического нуля.

Вышеупомянутые драйверы и импедансы могут быть выполнены в виде дискретных компонентов или как часть схемного модуля, или специализированной интегральной схемы

(специализированной ИС), которая служит в качестве экономически более эффективного решения кодера или декодера.

Легко можно видеть, что питание передается на устройство клиента, или дисплей, от устройства хоста, используя сигналы, обозначенные HOST_Pwr и HOST_Gnd, по паре проводников. Часть HOST_Gnd сигнала служит в качестве базового заземления и обратного тракта или сигнала источника питания для устройства клиента. Сигнал HOST_Pwr служит в качестве источника питания устройства клиента, который возбуждается устройством хоста. В примерной конфигурации для применений с низким потреблением мощности устройство клиента может потреблять ток до 500 мА. Сигнал HOST_Pwr может подаваться от портативных источников питания, таких как, но не ограничиваясь ими, аккумуляторная батарея или блок батарей ионно-литиевого типа, постоянно находящийся в устройстве хоста, и значение его может находиться в диапазоне от 3,2 до 4,3 В относительно HOST_Gnd.

VII. Временные характеристики

A. Обзор

Этапы и уровни сигналов, используемые для перехода в состояние «спячки» (не запрашивается, нежелательно или не требуется обслуживание) и обеспечения обслуживания клиента с хоста, или хостом, или клиентом инициированное, изображены на фиг.43а, 43b и 43с соответственно. На фиг.43а, 43b и 43с первая часть изображенных сигналов показывает Пакет закрытия линии передачи данных, пересылаемый с хоста, и линия данных затем возбуждается состоянием логического нуля, используя высокоимпедансную цепь смещения. Данные не передаются клиентом или хостом, в котором его драйвер выключен. В нижней части можно видеть группу строб-импульсов на линии сигнала MDDI_Stb, так как MDDI_Stb активен во время Пакета закрытия линии передачи данных. Когда этот пакет завершается, логический уровень меняется на нуль, когда хост возбуждает цепь смещения и логику в нуль. Это представляет завершение пересылки последнего сигнала или обслуживания с хоста и могло произойти в любой момент времени в прошлом, и оно включено для того, чтобы показать предшествующую приостановку обслуживания и состояние сигналов перед началом обслуживания. Если потребуется, то такой сигнал может быть послан просто для сброса линии обмена данными в надлежащее состояние без «известного» предшествующего обмена данными, предпринятого этим устройством хоста.

Как показано на фиг.43а и описано для Пакета закрытия линии передачи данных выше в состоянии «спячки» с низким потреблением мощности, драйвер MDDI_Data0 выключен в высокоимпедансное состояние, начиная после 16-го - 48-го цикла или импульса MDDI_Stb после последнего бита поля Всех нулей в Пакете закрытия линии передачи данных. Для линий передачи данных Типа-2, Типа-3 или Типа-4 сигналы MDDI_Data1 - MDDI_DataPwr7 также переводятся в высокоимпедансное состояние одновременно с выключением драйвера MDDI_Data0. Как описано при определении поля Всех нулей, MDDI_Stb переключается в течение 64 циклов (или как требуется для разработки системы) после ССБ поля ЦИК Пакета закрытия линии передачи данных, чтобы предоставить возможность завершить обработку клиентом и способствовать правильному отключению контроллера клиента. Один цикл представляет собой переход с низкого на высокий уровень, за которым следует переход с высокого на низкий уровень, или переход с высокого на низкий уровень, за которым следует переход с низкого на высокий уровень. После посылки поля Всех нулей драйверы MDDI_Stb и MDDI_Data0 в хосте выключаются, и хост переходит в состояние «спячки» с низким потреблением мощности. После некоторого периода времени хост начинает последовательность повторного запуска линии передачи данных, как показано на фиг.43b и 43с, посредством включения линий или выходов драйверов MDDI_Data0 и MDDI_Stb, и начинает переключать MDDI_Stb как часть иницируемого или хостом, или клиентом запроса пробуждения.

Как показано на фиг.43b, после того как пройдет некоторое время с выключенными сигнальными выходами драйверов для MDDI_Data0 и MDDI_Stb, хост иницирует

обслуживание или пробуждение из состояния «спячки» посредством включения его драйвера MDDI_Stb в течение периода времени, обозначенного $t_{\text{stb-dagta-enbl}}$, в течение которого линия возбуждается уровнем логического нуля до тех пор, пока она не будет полностью включена, и затем включения его драйвера MDDI_Data0. Хост удерживает

5 MDDI_Stb на уровне логического нуля, после того как MDDI_Data0 достигает высокого уровня или уровня логической единицы, что происходит через период времени, обозначенный $t_{\text{client-startup}}$. В конце периода $t_{\text{client-startup}}$ хост затем переключает сигнал или линию MDDI_Stb. Хост возбуждает линию MDDI_Data0 высоким уровнем, уровнем логической единицы, в то время как клиент не возбуждает MDDI_Data0 в течение периода, обозначенного $t_{\text{restart-high}}$, и затем возбуждает линию MDDI_Data0 низким уровнем, или

10 уровнем логического нуля в течение периода, обозначенного $t_{\text{restart-low}}$. После этого начинается первый прямой трафик с Пакета заголовка подкадра и затем пересылаются пакеты прямого трафика. Сигнал MDDI_Stb является активным во время периода $t_{\text{restart-low}}$ и последующего Пакета заголовка подкадра.

15 Как показано на фиг.43с, после того как пройдет некоторое время с выключенным сигнальным выходом от драйверов для MDDI_Data0 и MDDI_Stb, клиент инициирует запрос на обслуживание или пробуждение из состояния «спячки» посредством включения смещения в приемнике или выходном сигнале MDDI_Stb в течение периода времени, обозначенного $t_{\text{stb-dagta-enbl}}$, как описано выше, до того как хост включит свой драйвер MDDI_Stb. Затем клиент включает свой драйвер MDDI_Data0 в течение периода времени, обозначенного $t_{\text{host-detect}}$, во время которого линия возбуждается уровнем логического нуля, до того как хост начнет переключение MDDI_Stb.

Проходит или может потребоваться некоторое количество времени, перед тем как хост обнаружит запрос во время $t_{\text{host-detect}}$, после которого хост отвечает удержанием MDDI_Stb

25 на уровне логического нуля в течение периода, обозначенного $t_{\text{stb-startup}}$, до того как хост начнет переключение MDDI_Stb последовательностью запуска линии передачи данных посредством возбуждения MDDI_Data0 уровнем логической единицы или высоким уровнем в течение периода $t_{\text{restart-high}}$. Когда клиент распознает первый импульс на MDDI_Stb, он выключает смещение на своем приемнике MDDI_Stb. Клиент продолжает возбуждать

30 MDDI_Data0 уровнем логической единицы в течение периода, обозначенного $t_{\text{client-detect}}$, до тех пор пока он не обнаружит, что хост возбуждает линию. В этот момент клиент прекращает установление запроса и выключает свой драйвер MDDI_Data0, так что выходной сигнал с клиента переходит снова на уровень логического нуля, и хост возбуждает MDDI_Data0. Как и раньше, хост продолжает возбуждать MDDI_Data0 уровнем логической единицы в течение периода $t_{\text{restart-high}}$ и затем возбуждает линию MDDI_Data0

35 низким уровнем в течение периода $t_{\text{restart-low}}$, после которого начинается первый прямой трафик с Пакета заголовка подкадра. Сигнал MDDI_Stb является активным во время периода $t_{\text{restart-low}}$ и последующего Пакета заголовка подкадра.

В таблице VIII приведены характерные времена или периоды обработки для продолжительности различных периодов, описанных выше, и зависимость с примерными минимальными и максимальными скоростями передачи данных, где:

$$t_{\text{bit}} = \frac{1}{\text{Link_Data_Rate}}, \text{ где Link_Data_Rate представляет собой скорость}$$

передачи битов одной пары данных.

45

Таблица VIII					
Параметр	Описание	Мин.	Тип.	Макс.	Единицы
1/1 _{BIT} -min-perf	Скорость передачи данных по линии передачи данных для устройства с минимальными рабочими характеристиками	0,001		1,1	Мбит/с
1/1 _{BIT} -max-perf	Максимальный диапазон скорости передачи данных по линии передачи данных для устройства, внешний	0,001		400	Мбит/с
1/1 _{BIT} -max-perf	Максимальный диапазон скорости передачи данных по линии передачи данных для устройства, внутренний	0,001		550	Мбит/с
	Скорость передачи данных по обратной линии передачи данных	0,0005		50	Мбит/с

50

5	t_{bit}	Длительность одного бита данных прямой линии передачи данных, внешний режим	2,5		10^6	нс
	t_{bit}	Длительность одного бита данных прямой линии передачи данных, внутренний режим	1,8		10^6	нс
	$t_{restart-high}$	Длительность импульса с высоким уровнем повторного запуска линии передачи данных хоста	140	150	160	циклов Stb
	$t_{restart-low}$	Длительность импульса с низким уровнем повторного запуска линии передачи данных хоста	50	50	50	циклов Stb
	$t_{stb-data-enabl}$	Последовательность повторного запуска линии передачи данных от полного включения MDDI_Stb до включения MDDI_Data0	0			мкс
	$t_{client-startup}$	Время, необходимое хосту для удержания MDDI_Stb на уровне логического нуля, после того как MDDI_Data0 достигает логического высокого уровня	200			пс
	$t_{host-detect}$	Время от высокого уровня MDDI_Data0 до переключения MDDI_Stb	0		1000	мкс
	$t_{client-detect}$	Время, необходимое клиенту для обнаружения устройства с рабочими характеристиками MDDI_Data0 на логическом высоком уровне	60		80	циклов Stb
10	$t_{stb-startup}$	Время, необходимое хосту для удержания MDDI_Stb на уровне логического нуля, перед тем как хост начнет переключение MDDI_Stb	200			нс

Специалисту в данной области техники хорошо известно, что общеизвестны функции индивидуальных элементов, изображенных на фиг.41 и 42, и функция элементов на фиг.42 подтверждается временной диаграммой на фиг.43а, 43b и 43с. Подробности о последовательных согласованных нагрузках и резисторах состояния «спячки», которые показаны на фиг.42, были опущены на фиг.41, потому что эта информация необязательна для описания того, как выполнить кодирование Data-Strobe и восстановить из них тактовые импульсы.

В. Временные соотношения данных и строб-сигналов на прямой линии передачи данных

Характеристики переключения для пересылки данных по прямой линии передачи данных с выхода драйвера хоста приведены в таблице IX-1. В таблице IX представлена табличная форма требуемых минимума и максимума относительно типовых длительностей времени для некоторых происходящих переходов сигналов. Например, типовая продолжительность времени для перехода с начала до конца значения данных (выводной сигнал «0» или «1»), переход Data0-Data0, обозначенный $t_{tdd-(host-output)}$, равен t_{tbit} , тогда как минимальное время составляет примерно $t_{tbit}-0,5$ нс, и максимум составляет примерно $t_{tbit}+0,5$ нс.

Относительное расстояние между переходами на Data0, других линиях данных (DataX) и строб-линиях (Stb) показано на фиг.44, где показаны переходы Data0 - Strobe, Strobe - Strobe, Strobe - Data0, Data0 - non-Data0, non-Data0 - non-Data0, non-Data0 - Strobe и Strobe - non-Data0, которые упоминаются как $t_{tds-(host-output)}$, $t_{tss-(host-output)}$, $t_{tsd-(host-output)}$, $t_{tddx-(host-output)}$, $t_{tdxdx-(host-output)}$, $t_{tdxs-(host-output)}$ и $t_{tsdx-(host-output)}$ СООТВЕТСТВЕННО.

Таблица IX-1					
Параметр	Описание	Мин.	Тип.	Макс.	Единицы
$t_{tdd-(host-output)}$	Переход Data0 - Data0	$t_{tbit}-0,5$	t_{tbit}	$t_{tbit}+0,5$	нс
$t_{tds-(host-output)}$	Переход Data0 - Strobe	$t_{tbit}-0,8$	t_{tbit}	$t_{tbit}+0,8$	нс
$t_{tss-(host-output)}$	Переход Strobe - Strobe	$t_{tbit}-0,5$	t_{tbit}	$t_{tbit}+0,5$	нс
$t_{tsd-(host-output)}$	Переход Strobe - Data0	$t_{tbit}-0,8$	t_{tbit}	$t_{tbit}+0,8$	нс
$t_{tddx-(host-output)}$	Переход Data0 - non-Data0		t_{tbit}		нс
$t_{tdxdx-(host-output)}$	Переход non-Data0 - non-Data0	$t_{tbit}-0,5$	t_{tbit}	$t_{tbit}+0,5$	нс
$t_{tdxs-(host-output)}$	Переход non-Data0 - Strobe		t_{tbit}		нс
$t_{tsdx-(host-output)}$	Переход Strobe - non-Data0		t_{tbit}		нс

Типовые временные требования ЦИМД для входа приемника клиента для одинаковых сигналов, пересылающих данные по прямой линии передачи данных, приведены в таблице IX-2. Так как описываются одинаковые сигналы, но задержанные во времени, то не требуется новая фигура для иллюстрации характеристик сигнала или значения соответствующих обозначений, что понятно для специалистов в данной области техники.

Таблица IX-2					
Параметр	Описание	Мин.	Тип.	Макс.	Единицы
$t_{tdd-(client-input)}$	Переход Data0 - Data0	$t_{tbit}-1,0$	t_{tbit}	$t_{tbit}+1,0$	нс
$t_{tds-(client-input)}$	Переход Data0 - Strobe	$t_{tbit}-1,5$	t_{tbit}	$t_{tbit}+1,5$	нс
$t_{tss-(client-input)}$	Переход Strobe - Strobe	$t_{tbit}-1,0$	t_{tbit}	$t_{tbit}+1,0$	нс

$t_{sd}-(client-input)$	Переход Strobe - Data0	$t_{bit}-1,5$	t_{bit}	$t_{bit}+1,5$	нс
$t_{ddx}-(host-output)$	Переход Data0 - non-Data0		t_{bit}		нс
$t_{dxdx}-(host-output)$	Переход non-Data0 - non-Data0		t_{bit}		нс
$t_{dxs}-(host-output)$	Переход non-Data0 - Strobe		t_{bit}		нс
$t_{sdx}-(host-output)$	Переход Strobe - non-Data0		t_{bit}		нс

На фиг.45 и 46 изображено присутствие задержки в ответе, которая может иметь место, когда хост выключает или включает драйвер хоста соответственно. В случае хоста, направляющего некоторые пакеты, такие как Пакет инкапсуляции обратной линии передачи данных или Пакет измерения задержки на двойное прохождение, хост выключает линейный драйвер после передачи требуемых пакетов, таких как пакеты ЦИК параметров, Выравнивания строка и Всех нулей, изображенных на фиг.45, которые были пересланы. Однако, как показано на фиг.45, состояние линии необязательно переключается мгновенно с «0» на требуемое более высокое значение, хотя это потенциально достижимо с некоторым присутствующим управлением или схемными элементами, но занимает для ответа некоторый период времени, называемый периодом Задержки выключения драйвера хоста. Хотя он может происходить фактически мгновенно, так что этот период времени будет составлять 0 наносекунд (нс) по продолжительности, но он может легко растянуться на некоторый более продолжительный период, причем 10 нс является требуемой максимальной длительностью периода, который происходит в течение периодов пакетов Защитного интервала 1 или Реверсирования передачи 1.

Как показано на фиг.46, видно, что уровень сигнала претерпевает изменение, когда Драйвер хоста включается для пересылки пакета, такого как Пакет инкапсуляции обратной линии передачи данных или Пакет измерения задержки на двойное прохождение. В данном случае после периодов времени пакета Защитного интервала 2 или Реверсирования передачи 2 драйвер хоста включается и начинает возбуждать уровень, в данном случае «0», причем это значение приближается или достигается в течение периода времени, называемого периодом Задержки на включение драйвера хоста, который происходит в течение периода Повторного включения драйвера перед посылкой первого пакета.

Аналогичный процесс происходит для драйверов и пересылки сигналов для устройства клиента, в данном случае дисплея. Общие рекомендации по длительности этих периодов и их соответствующих зависимостей приведены ниже в таблице X.

Таблица X			
Описание	Мин.	Макс.	Единицы
Задержка на выключение драйвера хоста	0	10	нс
Задержка на включение драйвера хоста	0	2,0	нс
Задержка на выключение драйвера дисплея	0	10	нс
Задержка на включение драйвера дисплея	0	2,0	нс

С. Время включения и выключения выходов хоста и клиента

Характеристики переключения и относительные временные зависимости для времени или операций включения и выключения выходов Хоста и Клиента относительно структуры и периода Пакета инкапсуляции обратной линии передачи данных показаны на фиг.48. Функции или операции выхода драйвера обозначаются как: $t_{host-enable}$ для времени включения выхода Хоста, $t_{host-disable}$ для времени выключения выхода Хоста, $t_{client-enable}$ для времени включения выхода Клиента и $t_{client-disable}$ для времени выключения выхода Клиента. Типовые времена для некоторых переходов сигнала описываются ниже. Минимальный период для этих операций равен нулю наносекунд, причем типовые или максимальные значения определяются из разработки системы, использующей интерфейс, возможно порядка 8 наносекунд, или более.

Общие рекомендации по длительности этих периодов (времени включения/выключения клиента и хоста) и их соответствующих зависимостей показаны в таблице XI ниже.

Таблица XI			
Параметр	Описание	Мин.	Тип. Макс. Единицы

5	t _{host-enable}	Время включения выхода хоста	0		24.t BIT	нс
	t _{host-disable}	Время выключения выхода хоста, вся продолжительность поля Реверсирования передачи 1	0		24.t BIT	нс
	t _{client-enable}	Время включения выхода клиента, вся продолжительность поля Реверсирования передачи 1	0		24.t BIT	нс
	t _{client-disable}	Время выключения выхода клиента, измеренное от окончания последнего бита поля Реверсирования передачи 2	0		24.t BIT	нс

VIII. Реализация управления линией передачи данных (принцип действия контроллера линии передачи данных)

A. Процессор пакетов на конечном автомате

Пакеты, пересылаемые по линии передачи данных ЦИМД, отправляются очень быстро, обычно со скоростью порядка 300 Мбит/с или более, такой как 400 Мбит/с, хотя по необходимости, конечно, обеспечиваются меньшие скорости передачи. Этот тип скорости шины или линии пересылки данных слишком большой для управления серийно выпускающимися в настоящее время (экономичными) микропроцессорами общего назначения или им подобными. Поэтому практической реализацией для осуществления пересылки сигнала этого типа является использование программируемого конечного автомата для синтаксического анализа потока входных пакетов для получения пакетов, которые пересылаются или перенаправляются на соответствующую аудиовизуальную подсистему, для которой они предназначены. Такие устройство общеизвестны и используют схемы, как правило, предназначенные для ограниченного количества операций, функций или состояний для достижения работы с требуемой высокой скоростью или с очень высокой скоростью.

Контроллеры, процессоры или обрабатывающие элементы общего назначения могут использоваться для более соответствующего действия или управления некоторой информацией, такой как пакеты управления или состояния, у которых более низкие требования к скорости. Когда принимаются такие пакеты (пакеты управления, состояния или другие предварительно определенные пакеты), конечный автомат должен пропустить их через буфер данных или подобный обрабатывающий элемент на процессор общего назначения, так что на пакеты может быть оказано воздействие для получения требуемого результата (эффекта), тогда как аудио- и визуальные пакеты пересылаются по их соответствующему назначению для совершения действия. Если будущие микропроцессоры или другие контроллеры, процессоры или обрабатывающие элементы общего назначения будут производиться для достижения возможностей обработки с более высокими скоростями передачи данных, тогда состояния или конечный автомат, описанный ниже, также могут быть реализованы с использованием программного управления такими устройствами обычно в качестве программ, хранимых на запоминающем элементе или носителе данных.

Функция процессора общего назначения может быть реализована в некоторых вариантах осуществления, воспользовавшись вычислительной мощностью, или избыточными циклами, доступными для микропроцессоров (ЦП) в компьютерных применениях, или контроллеров, процессоров, цифровых процессоров сигналов (ЦПС) или специализированных схем, или специализированных ИС, находящихся в беспроводных устройствах, почти таким же образом, как и некоторые модемы или графические процессоры используют вычислительную мощность ЦП в компьютерах для выполнения некоторых функций и снижения сложности и стоимости аппаратных средств. Однако это разделение или использование циклов может оказать отрицательное воздействие на скорость обработки, временные соотношения или общую работу таких элементов, поэтому для этой общей обработки во многих применениях предпочтительны специализированные схемы или элементы.

Для того чтобы наблюдать данные изображения на дисплее (микродисплее) или надежно принимать все пакеты, посылаемые устройством хоста, обработка сигнала

клиента синхронизируется с временными соотношениями канала прямой линии передачи данных. Т.е. сигналы, поступающие на клиент и схемы клиента, должны по существу синхронизироваться во времени, чтобы происходила надлежащая обработка сигнала.

Высокоуровневая диаграмма состояний, достигаемых сигналом для одного варианта осуществления, представлена на иллюстрации на фиг.49. На фиг.49 возможные «состояния» синхронизации прямой линии передачи данных для конечного автомата 4900 показаны сгруппированными в виде одного Состояния 4904 асинхронных кадров, двух Приобретающих синхронизацию состояний 4902 и 4906 и трех Состояний 4908, 4910 и 4912 в синхронизме.

Как показано посредством начального этапа или состояния 4902, дисплей или клиент, такой как устройство представления, начинает работу в предварительно выбранном состоянии «без синхронизации» и производит поиск служебного слова в первом пакете заголовка подкадра, который обнаружен. Необходимо заметить, что это состояние без синхронизации представляет установку минимального обмена данными или установку «возврата в исходное состояние», при которой выбирается интерфейс Типа 1. Когда во время поиска обнаружено служебное слово, то клиент сохраняет поле длины подкадра. Нет проверки битов ЦИК для обработки этого первого кадра, или пока не будет достигнута синхронизация. Если эта длина подкадра равна нулю, то тогда обработка состояния синхронизации переходит соответствующим образом на состояние 4904, обозначенное в данном случае как состояние «асинхронных кадров», что указывает, что синхронизация еще не была достигнута. Этот этап в обработке обозначается как имеющий встречаемое cond 3, или условие 3, на фиг.49. В противном случае, если длина кадра больше нуля, то тогда обработка состояний синхронизации переходит к состоянию 4906, где состояние интерфейса устанавливается как «найден один синхронизированный кадр». Этот этап в обработке обозначается как встречающееся cond 5, или условие 5, на фиг.49. Кроме того, если конечный автомат определяет пакет заголовка кадра и успешное определение ЦИК для длины кадра, большей нуля, то обработка переходит к состоянию «найден один синхронизированный кадр». Это обозначается как удовлетворяющее cond 6, или условие 6, на фиг.49.

В любой ситуации, в которой система находится в состоянии, отличном от состояния «без синхронизации», если обнаруживается пакет с успешным результатом ЦИК, то тогда состояние интерфейса меняется на состояние 4908 «в синхронизме». Этот этап в обработке обозначается как имеющий встречаемое cond 1, или условие 1, на фиг.49. С другой стороны, если ни в каком пакете ЦИК не является правильным, тогда обработка состояний синхронизации переходит или возвращается в состояние 4902 интерфейса, состояние «Кадр без синхронизации». Эта часть обработки обозначается как встречающееся cond 2, или условие 2, на диаграмме состояний на фиг.49.

В. Время приобретения синхронизации

Интерфейс может конфигурироваться для обеспечения некоторого количества «ошибок синхронизации» перед принятием решения, что синхронизация потеряна, и возвратом в состояние «Кадра без синхронизации». На фиг.49, если конечный автомат достиг «Состояния в синхронизме», и не обнаружено ошибок, то он постоянно встречает результат cond 1 и остается в состоянии «В синхронизме». Однако если обнаружен один результат cond 2, то обработка меняет состояние на состояние 4910 «одна ошибка синхронизации». В этот момент, если обработка приводит к обнаружению другого результата cond 1, тогда конечный автомат возвращается в состояние «в синхронизме», в противном случае он встречает другой результат cond 2 и переходит в состояние 4912 «Две ошибки синхронизации». Снова, если происходит cond 1, обработка возвращает конечный автомат в состояние «в синхронизме». В противном случае, встречается другое cond 2 и конечный автомат возвращается в состояние «без синхронизации». Также понятно, что, если интерфейс встречает «пакет закрытия линии передачи данных», тогда это вызывает завершение линией передачи данных пересылок данных и возврат в состояние «кадр без синхронизации», так как нет ничего с чем синхронизироваться, что

упоминается как удовлетворяющее cond 4, или условие 4, на диаграмме состояний на фиг.49.

Понятно, что возможно существование повторяющейся «ложной копии» служебного слова, которое может появляться в некотором фиксированном месте в подкадре. В этом случае весьма маловероятно, что конечный автомат будет синхронизироваться с подкадром, так как ЦИК Пакета заголовка подкадра также должен быть действительным при обработке, чтобы обработка ЦИМД перешла в состояние «В синхронизме».

Длина подкадра в Пакете заголовка подкадра может быть установлена в нуль для указания, что хост будет передавать только один подкадр перед закрытием линии передачи данных, и ЦИМД переводится или конфигурируется в состояние «спячки» незанятости. В этом случае клиент должен немедленно принять пакеты по прямой линии передачи данных после обнаружения Пакета заголовков подкадра, потому что только один подкадр посылается перед переходами линии передачи данных в состояние незанятости. При нормальной или обычной работе длина подкадра не является нулевой, и клиент обрабатывает только пакеты прямой линии передачи данных, тогда как интерфейс находится в тех состояниях, которые показаны вместе как состояния «В синхронизме» на фиг.49.

Устройство клиента во внешнем режиме может подключаться к хосту, в то время как хост уже передает последовательность данных по прямой линии передачи данных. В данной ситуации клиент должен синхронизироваться с хостом. Время, необходимое для синхронизации клиента с сигналом на прямой линии передачи данных, является переменным в зависимости от размера подкадра и скорости передачи по прямой линии передачи данных. Вероятность обнаружения «ложной копии» служебного слова как части случайных или более случайных данных на прямой линии передачи данных больше тогда, когда больше размер подкадра. Одновременно хуже способность восстановления при ложном обнаружении, и продолжительнее время, которое необходимо для его выполнения при более низкой скорости передачи данных по прямой линии передачи данных.

Для одного или нескольких вариантов осуществления рекомендуется или понятно, что хост ЦИМД должен выполнять некоторые дополнительные этапы для гарантирования того, что обратная линия передачи данных ЦИМД является стабильной, перед тем как она прекращает передачу по прямой линии передачи данных для перехода в режим с низким потреблением мощности или закрытия полностью линии передачи данных.

Одной проблемой, которая может иметь место, является то, что, если хост использует неправильное измерение значения задержки на двойное прохождение, то это может вызывать то, что все принятые в последствие передачи обратных данных от клиента являются неуспешными, даже если прямая линия передачи данных находится, по-видимому, в отличном состоянии. Это может происходить, если хост делает попытку послать Пакет измерения задержки на двойное прохождение, когда клиент не находится в синхронизме с прямой линией передачи данных, или из-за предельного изменения окружающей температуры, которое вызывает соответствующее большое изменение задержки на распространение дифференциальных драйверов и приемников, которое оказывают влияние на задержку на двойное прохождение. Переменяющаяся неисправность кабеля или контакта соединителя также может вызывать то, что клиент будет временно терять синхронизацию и затем восстанавливать синхронизацию, причем в течение этого времени он может не принимать Пакет измерения задержки на двойное прохождение. Последующие пакеты обратной линии передачи данных будут не в состоянии правильно декодироваться хостом.

Другим типом проблемы, которая может иметь место, является то, что, если клиент временно теряет синхронизацию, хост посылает Пакет закрытия линии передачи данных до того времени, когда клиент будет способен восстановить синхронизацию. Хост будет находиться в состоянии «спячки», тогда как клиент не сможет войти в состояние «спячки», так как он не принял Пакет закрытия линии передачи данных и у него нет тактовых импульсов, так как линия передачи данных находится в состоянии «спячки».

Одним методом или вариантом осуществления, полезным для решения таких проблем, является обеспечение того, что хост гарантирует то, что клиент находится в синхронизме с прямой линией передачи данных перед переводом линии передачи данных в состояние «спячки». Если хост ЦИМД не может это выполнить или не имеет такой

5 возможности, такой что, когда он теряет питание или линия передачи данных внезапно разрывается или выходит из строя из-за разъединения, обрыва или отсоединения кабеля, проводника или соединителя, происходящих во время работы, тогда хост должен сначала попытаться обеспечить, чтобы клиент был в синхронизме перед началом процесса измерения задержки на двойное прохождение или посылкой пакета Инкапсуляции обратной

10 линии передачи данных.

Хост может исследовать поле Числа ошибок ЦИК в пакете Запроса клиента и состояния, посылаемое клиентом для определения целостности прямой линии передачи данных. Этот пакет запрашивается хостом у клиента. Однако в случае большой неисправности или разрыва линии передачи данных наиболее вероятно, что этот запрос окажется без ответа,

15 так как клиент не будет иметь возможность правильно декодировать пакет и даже, может быть, вообще его принять. Запрос в отношении Числа ошибок ЦИК, используя Пакет запроса клиента и состояния, посылаемый в Пакете инкапсуляции обратной линии передачи данных, действует в качестве первой проверки на целостность, своего рода первой линией обороны. Кроме того, хост может посылать Пакет измерения задержки на двойное прохождение для подтверждения, является ли действительным или нет

20 предположение о том, что клиент вышел из синхронизма. Если клиент не отвечает на Пакет измерения задержки на двойное прохождение, хост делает заключение, что клиент находится вне синхронизации, и может тогда начать процесс возвращения его в синхронизм.

Если хост делает заключение, что более чем вероятно, что клиент потерял синхронизацию с прямой линией передачи данных, он ожидает до следующего заголовка подкадра перед выполнением попытки послать любые пакеты кроме пакетов заполнителя. Это делается для того, чтобы предоставить клиенту достаточное время для обнаружения или поиска одного служебного слова, содержащегося в пакете заголовка подкадра. После

30 этого хост может предположить, что клиент выполнил сброс самого себя, так как он не обнаружил служебное слово в правильном расположении. В этот момент хост может подавать Пакет измерения задержки на двойное прохождение после пакета заголовка подкадра. Если клиент все же не отвечает правильно на Пакет измерения задержки на двойное прохождение, хост может повторить процесс ресинхронизации. Правильным

35 ответом является ответ, в котором клиент посылает заданную последовательность обратно хосту в Периоде измерения Пакета измерения задержки на двойное прохождение. Если эта последовательность не принимается, тогда завершаются неуспешно попытки приема обратных данных в Пакете инкапсуляции обратной линии передачи данных. Продолжающаяся неисправность такого рода может указывать на некоторую другую

40 системную ошибку, на которую необходимо реагировать некоторым другим образом, и она не является частью синхронизации линии передачи данных в этот момент.

Однако, если после успешного Пакета измерения задержки на двойное прохождение хост все же замечает искаженные данные или отсутствие ответа в Пакетах инкапсуляции обратной линии передачи данных, он должен подтвердить, что отсчет обратных данных

45 правильный посредством повторной посылки Пакета измерения задержки на двойное прохождение. Если это не является успешным после ряда попыток, рекомендуется для одного варианта осуществления, чтобы хост снизил скорость передачи обратных данных посредством повышения значения делителя обратной скорости передачи.

Хост должен выполнить этапы Обнаружения неисправности линии передачи данных и, возможно, Ресинхронизации линии передачи данных, описанные выше, перед переводом

50 линии передачи данных ЦИМД в состояние «спячки». Это, как правило, гарантирует то, что является успешным Пакет измерения задержки на двойное прохождение, посланный тогда, когда позже осуществляется повторный запуск линии передачи данных. Если у

хоста нет причины подозревать неисправность линии передачи данных, и правильный ответ на Пакет инкапсуляции обратной линии передачи данных и нулевые ошибки ЦИК по прямой линии передачи данных сообщаются клиентом, хост может предположить, что все работает или функционирует соответствующим или надлежащим образом (например, нет
5 неисправности линии передачи данных), и продолжить процесс снижения потребляемой мощности/«спячки».

Другим образом, при помощи которого хост может проверить синхронизацию, является тот, что хост посылает Пакет измерения задержки на двойное прохождение и подтверждает
10 надлежащий ответ от клиента. Если хостом принимается надлежащий ответ, он может корректно предположить, что клиент успешно интерпретирует пакеты прямой линии передачи данных.

С. Инициализация

Как указано выше, во время «запуска» хост конфигурирует прямую линию передачи данных для работы при минимально требуемой, или необходимой, скорости передачи
15 данных 1 Мбит/с или при более низкой и конфигурирует длину подкадра и скорость передачи медиакадров соответствующим образом для данного применения. Т.е. как прямая, так и обратная линия передачи данных начинают работу с использованием интерфейса Типа 1. Эти параметры, как правило, предполагается использовать только временно, пока хост определяет возможности или требуемую конфигурацию для дисплея
20 клиента (или другого типа устройства клиента). Хост посылает или пересылает Пакет заголовка подкадра по прямой линии передачи данных, за которым следует Пакет инкапсуляции обратной линии передачи данных, у которого бит «0» Флагов запроса установлен в значение единицы (1) для запроса, чтобы дисплей или клиент отвечал Пакетом возможностей клиента. Если дисплей достигает синхронизации по (или с) прямой
25 линии передачи данных, он посылает Пакет возможностей клиента и Пакет запроса клиента и состояния по обратной линии или каналу передачи данных.

Хост исследует содержимое Пакета возможностей клиента с целью определения, как реконфигурировать линию передачи данных для оптимального или требуемого уровня рабочих характеристик. Хост исследует поля Версии протокола и Минимальной версии
30 протокола для подтверждения, что хост и клиент используют версии протокола, которые являются совместимыми друг с другом. Версии протоколов, как правило, остаются в качестве первых двух параметров Пакета возможностей клиента, так что совместимость может определяться даже тогда, когда другие элементы протокола могли бы быть несовместимыми или полностью определяемыми как несовместимые.

Во внутреннем режиме хост может иметь сведения о параметрах клиента заранее без
35 необходимости приема Пакета возможностей клиента. Линия передачи данных может запускаться с любой скоростью передачи данных, с которой хост и клиент оба могут работать. Во многих вариантах осуществления разработчик системы, наиболее вероятно, выбирает запуск линии передачи данных с максимально достижимой скоростью передачи
40 данных, чтобы ускорить пересылку данных, однако это не требуется и может не использоваться во многих ситуациях. Для работы во внутреннем режиме частота строб-импульсов, используемая во время повторного запуска линии передачи данных из последовательности «спячки», обычно согласуется с этой требуемой скоростью передачи данных.

D. Обработка ЦИК

Для всех типов пакетов конечный автомат процессора пакетов обеспечивает, что устройство проверки ЦИК управляется соответствующим или надлежащим образом. Он также дает приращение счетчику ошибок ЦИК, когда сравнение ЦИК приводит к
45 обнаруживаемым одной или несколькими ошибкам, и он сбрасывает счетчик ЦИК в начале каждого обрабатываемого подкадра.

E. Альтернативная проверка потери синхронизации

Хотя вышеупомянутые последовательности этапов или состояний работают для получения более высоких скоростей передачи данных или производительности, заявители

обнаружили, что альтернативное упорядочение или изменение в условиях, которые клиент использует для объявления, что существует потеря синхронизации с хостом, могут эффективно использоваться для достижения еще более высоких скоростей передачи данных или пропускной способности. Новый вариант осуществления изобретения имеет
 5 такую же базовую структуру, но с измененными условиями для изменения состояний. Кроме того, реализован новый счетчик, способствующий выполнению проверок в отношении синхронизации подкадров. Эти этапы и условия представлены относительно фиг.63, которая иллюстрирует последовательности состояний и условий, полезных при
 10 установлении операций способа или конечного автомата. Для ясности показаны только части «Приобретающих синхронизацию состояний» и «Состояний в синхронизме». Кроме того, так как результирующие состояния по существу одинаковые, как и сам конечный автомат, они используют одинаковую нумерацию. Однако условия для смены состояний (и принцип действия конечного автомата) в некоторой степени изменяются, так что все перенумерованы для ясности между двумя фигурами (1, 2, 3, 4, 5 и 6 против 61, 62, 63,
 15 64 и 65) в качестве удобства при идентификации различий. Так как состояние Асинхронный кадр не рассматривается при данном описании, одно состояние (4904) и условие (6) больше не используются на фигуре.

На фиг.63 система или клиент (для отображения или презентации) начинают с конечным автоматом 5000 в предварительно выбранном состоянии 4902 «без синхронизации», как на
 20 фиг.49. Первое изменение состояния для изменения состояний от условия 4902 без синхронизации заключается в условии 64, которое представляет собой обнаружение комбинации синхронизации. Предполагая, что ЦИК заголовка подкадра также проходит по этому пакету (удовлетворяет условию 61), состояние конечного автомата процессора пакетов может быть изменено на состояние 4908 в синхронизме. Ошибка синхронизации,
 25 условие 62, вызывает переход конечного автомата в состояние 4910, и второе появление - в состояние 4912. Однако было обнаружено, что любой сбой ЦИК пакета ЦИМД вызывает выход конечного автомата из состояния 4908 в синхронизме в состояние 4910 одной ошибки синхронизации. Другой сбой ЦИК любого пакета ЦИМД вызывает переход в состояние 4912 двух ошибок синхронизации. Пакет, декодированный с
 30 правильным значением ЦИК, вызывает возврат конечного автомата в состояние 4908 в синхронизме.

Тем, что было изменено, является использование значения ЦИК или определения для «каждого» пакета. Т.е. просмотр конечным автоматом значения ЦИК для каждого пакета для определения потери синхронизации вместо просто наблюдения пакетов заголовка подкадра. В данной конфигурации или процессе потеря синхронизации не определяется с
 35 использованием служебного слова и просто значений ЦИК заголовка подкадра.

Эта новая реализация интерфейса дает возможность линии передачи данных ЦИМД значительно быстрее распознавать неисправности синхронизации и также, следовательно, быстрее восстанавливать из них.

40 Чтобы сделать данную систему более устойчивой к ошибкам, клиент также должен добавить или использовать счетчик подкадров. Клиент тогда проверяет присутствие служебного слова в момент времени, когда ожидается его появление или когда он происходит в сигнале. Если служебное слово не происходит в корректный момент времени, клиент может распознать, что произошел сбой синхронизации значительно быстрее, чем
 45 если бы ему пришлось ожидать продолжительности или периодов нескольких (в данном случае трех) пакетов, которые были больше продолжительности подкадра. Если тест на служебное слово указывает, что оно не присутствует, другими словами, что временные соотношения неправильные, тогда клиент может немедленно объявить о потере синхронизации линии передачи данных и перейти в состояние без синхронизации. Процесс проверки надлежащего присутствия служебного слова добавляет условие 65 (cond 65) к
 50 конечному автомату, сообщающее, что служебное слово неправильное. Если на клиенте ожидается прием пакета подкадра и он не согласуется, клиент может немедленно перейти в состояние 4902 без синхронизации, экономя дополнительное время, ожидая

многочисленных ошибок синхронизации (условие 62), обычно встречаемых, проходя через состояния 4910 и 4912.

Это изменение использует дополнительный счетчик или функцию подсчета в ядре клиента для подсчета длины подкадра. В одном варианте осуществления используется функция подсчета в обратном направлении, и пересылка любого пакета, который в настоящее время обрабатывался, прерывается для проверки служебного слова подкадра, если истекает срок в счетчике. Альтернативно счетчик может подсчитывать в прямом направлении, причем счет сравнивается с требуемым максимумом или конкретным требуемым значением, в этот момент текущий пакет проверяется. Этот процесс защищает клиента от декодирования пакетов, которые принимаются неправильно на клиенте с очень большими длинами пакетов. Если счетчику длины подкадра необходимо прервать некоторый другой пакет, который декодировался, может быть определена потеря синхронизации, так как пакет не должен пересекать границу подкадра.

IX. Обработка пакетов

Для описанного выше пакета каждого типа, который принимает конечный автомат, он выполняет конкретный этап или группу этапов по обработке для осуществления работы интерфейса. Пакеты прямой линии передачи данных, как правило, обрабатываются согласно примерной обработке, приведенной ниже в таблице XII.

Таблица XII	
Тип пакета	Реакция конечного автомата процессора пакетов
Заголовок подкадра (SH)	Подтверждает хороший пакет, захватывает поле длины подкадра и посылает параметры пакета на процессор общего назначения.
Заполнитель (F)	Игнорирует данные.
Видеопоток (VS)	Интерпретирует Дескриптор формата видеоданных и другие параметры, распаковывает упакованные данные пикселей, когда это необходимо, преобразует пиксели при помощи карты цветов, если это необходимо, и записывает данные пикселей в соответствующие места растрового изображения.
Аудиопоток (AS)	Посылает установку частоты аудиоотсчетов на тактовый генератор аудиоотсчетов, выделяет аудиоотсчеты заданного размера, распаковывает данные аудиоотсчетов, когда это необходимо, и направляет аудиоотсчеты на соответствующий стек обратного магазинного типа (COMT) аудиоотсчетов.
Карта цветов (CM)	Считывает параметры размера и смещения карты цветов и записывает данные карты цветов в ячейку память или запоминающего устройства карты цветов.
Инкапсуляция обратной линии передачи данных (REL)	Способствуетсылке пакетов в обратном направлении в соответствующий момент времени. Исследуются флаги обратной линии передачи данных, и по необходимости посылаются пакеты Возможностей клиента. Также посылаются пакеты Запроса клиента и состояния соответствующим образом.
Возможности клиента (CC)	Посылает пакет этого типа при запросе хостом, используя поле флагов обратной линии передачи данных Пакета инкапсуляции обратной линии передачи данных.
Клавиатура (K)	Пропускает эти пакеты на процессор общего назначения и с него, который участвует в обмене данными с устройством типа клавиатуры, если оно присутствует и требуется его использование.
Указательное устройство (PD)	Пропускает эти пакеты на процессор общего назначения и с него, который участвует в обмене данными с устройством указательного типа, если оно присутствует и требуется его использование.
Закрытие линии передачи данных (LS)	Регистрирует факт, что линия передачи данных закрывается, и информирует процессор общего назначения.
Запрос на обслуживание и состояние клиента (CSRS)	Посылает этот пакет в качестве первого пакета в пакете Инкапсуляции обратной линии передачи данных.
Пересылка битовых блоков (BPT)	Интерпретирует параметры пакета, такие как Дескриптор формата видеоданных, определяет, какие пиксели перемещать первыми, и перемещает пиксели в растровом изображении по необходимости.
Заливка растровой области (BAF)	Интерпретирует параметры пакета, преобразует пиксели при помощи карты цветов, если необходимо, и записывает данные пикселей в соответствующие места в растровом изображении.
Заливка растровым узором (BPF)	Интерпретирует параметры пакета, распаковывает упакованные данные пикселей, если необходимо, преобразует пиксели при помощи карты цветов, если необходимо, и записывает данные пикселей в соответствующие места в растровом изображении.
Канал линии обмена данными (CLC)	Посылает эти данные непосредственно на процессор общего назначения
Запрос на обслуживание клиента (CSR) во время «спячки»	Процессор общего назначения управляет низкоуровневыми функциями отправки запроса и со своей стороны обнаруживает состязание с повторным запуском линии передачи данных.
Запрос перехода на тип интерфейса (ITHR) и Подтверждение типа интерфейса (ITA)	Может пропускать эти пакеты на процессор общего назначения и с него. Логика для приема пакета этого типа и составления ответа с подтверждением по существу минимальна. Поэтому эта операция также может быть реализована в конечном автомате процессора пакетов. Результирующий переход происходит как действие низкоуровневого физического уровня, и маловероятно, чтобы оказывало влияние на функциональные возможности или функционирование процессора общего назначения.
Выполнение перехода на тип (PTH)	Может действовать на такие пакеты или непосредственно, или посредством пересылки их на процессор общего назначения, также посылая команду аппаратным средствам на выполнение изменения режима.

X. Снижение скорости передачи данных обратной линии передачи данных

Изобретателями было замечено, что некоторые параметры, используемые для контроллера линии передачи данных хоста, можно регулировать или конфигурировать определенным образом для достижения максимальной или более оптимизированной (масштабированной) скорости передачи данных по обратной линии передачи данных, что
 5 очень желательно. Например, в течение времени, используемого для пересылки поля Пакетов обратных данных Пакета инкапсуляции обратной линии передачи данных, пара сигналов MDDI_Stb переключается и создает периодический тактовый сигнал данных с половинной скоростью передачи данных прямой линии передачи данных. Это происходит, потому что контроллер линии передачи данных хоста генерирует сигнал MDDI_Stb, который
 10 соответствует сигналу MDDI_Data0, как если бы он посылал все нули. Сигнал MDDI_Stb пересылается с хоста на клиент, где он используется для генерирования тактового сигнала для пересылки данных обратной линии передачи данных с клиента, с которым обратные данные посылаются обратно на хост. На фиг.50 показана иллюстрация типовых значений задержки, которые имеют место при пересылке и обработке сигналов в прямом и
 15 обратном тракте в системе, использующей ЦИМД. На фиг.50 показана группа значений задержки: 1,5 нс, 8,0 нс, 2,5 нс, 2,0 нс, 1,0 нс, 1,5 нс, 8,0 нс и 2,5 нс около обрабатывающих частей для генерирования Stb+/-, пересылки клиенту по кабелю, приемника клиента, генерирования тактовых импульсов, тактирования сигнала, генерирования Data0+/-, пересылки на хост по кабелю и каскадов приемника хоста
 20 соответственно.

В зависимости от имеющей место скорости передачи данных по прямой линии передачи данных и задержек на обработку сигнала может потребоваться времени больше, чем один цикл сигнала MDDI_Stb для завершения этого эффекта «двойного прохождения» или группы событий, что приводит к расходованию нежелательного количества времени или
 25 циклов. Для решения этой проблемы Делитель обратной скорости передачи делает возможным, чтобы продолжительность одного битового интервала по обратной линии передачи данных охватывала многочисленные циклы сигнала MDDI_Stb. Это означает, что скорость передачи данных по обратной линии передачи данных меньше, чем скорость передачи по прямой линии передачи данных.

Необходимо заметить, что фактическая продолжительность задержек сигнала по интерфейсу может отличаться в зависимости от каждой заданной системы хост-клиент или используемых аппаратных средств. Хотя и не требуется, каждая система, как правило, может быть сделана так, что она будет работать лучше посредством использования Пакета измерения задержки на двойное прохождение для измерения фактической задержки в
 35 системе, так что Делитель обратной скорости передачи может устанавливаться на оптимальную величину. Хост может поддерживать или базовый отсчет данных, что проще, но работает при меньшей скорости, или улучшенный отсчет данных, который более сложный, но поддерживает более высокие скорости передачи обратных данных. Возможности клиента для поддержки обоих способов считаются одинаковыми.

Задержка на двойное прохождение измеряется посредством того, что хост посылает клиенту Пакет измерения задержки на двойное прохождение. Дисплей отвечает на этот пакет посылкой последовательности единиц обратно на хост внутри или в течение
 40 предварительно выбранного окна измерения в этом пакете, называемом полем Периода измерения. Подробные временные соотношения этого измерения были описаны ранее. Задержка на двойное прохождение используется для определения скорости, с которой может безопасно выполняться отсчет данных обратной линии передачи данных.

Измерение задержки на двойное прохождение состоит из определения, обнаружения или подсчета количества тактовых интервалов данных прямой линии передачи данных, происходящих между началом поля Периода измерения и началом периода времени, когда
 50 последовательность ответа 0xff, 0xff, 0x00 принимается обратно на хосте от клиента. Отметьте, что возможно, что ответ от клиента может приниматься за небольшую долю тактового периода прямой линии передачи данных до того, как число измерений собиралось получить приращение. Если эта неизменная величина используется для

вычисления Делителя обратной скорости передачи, она может вызвать ошибки битов на обратной линии передачи данных вследствие ненадежного отсчета данных. Пример этого случая изображен на фиг.51, где в графической форме изображены сигналы, представляющие MDDI_Data на хосте, MDDI_Stb на хосте, тактовые импульсы данных 5 прямой линии передачи данных внутри хоста и Число задержки. На фиг.51 последовательность ответа была принята от клиента за долю тактового периода прямой линии передачи данных до того, как Число задержки собиралось получить приращение с 6 до 7. Если задержка принимается равной 6, тогда хост будет производить отсчет 10 обратных данных как раз после битового перехода или, возможно, в середине битового перехода. Это может привести к ошибочному отсчету на хосте. По этой причине измеренная задержка обычно должна быть увеличена на единицу перед тем, как она будет использоваться для вычисления Делителя обратной скорости передачи.

Делитель обратной скорости передачи представляет собой число циклов MDDI_Stb, 15 которые хост должен ожидать перед отсчетом данных обратной линии передачи данных. Так как MDDI_Stb циклически повторяется с частотой, которая составляет половину скорости передачи по прямой линии передачи данных, то скорректированное измерение задержки на двойное прохождение необходимо разделить на два и затем округлить до 20 следующего целого числа. Эта зависимость, выраженная в виде формулы, следующая:

Округление До Следующего Целого Числа

$$\left(\frac{\text{задержка_на_двойное_прохождение} + 1}{2} \right)$$

Для данного примера она становится:
делитель_обратной_скорости_передачи =

Округление До Следующего Целого Числа $\left(\frac{6+1}{2} \right) = 4$

Если измерение задержки на двойное прохождение, использованное в этом примере, 30 составляло 7 против 6, то тогда Делитель обратной скорости передачи также был бы равен 4.

Отсчет данных обратной линии передачи данных выполняется хостом на переднем фронте Тактового импульса обратной линии передачи данных. Счетчик или аналогичная известная схема или устройство присутствует как на хосте, так и на клиенте (дисплее) для генерирования Тактового импульса обратной линии передачи данных. Счетчики 35 инициализируются, так что первый передний фронт Тактового импульса обратной линии передачи данных происходит в начале первого бита в поле Пакетов обратной линии передачи данных пакета Инкапсуляции обратной линии передачи данных. Это иллюстрируется для приведенного ниже примера на фиг.52A. Счетчики получают приращение на каждом переднем фронте сигнала MDDI_Stb, и количество подсчетов, 40 происходящих до тех пор, пока они не выполнят циклический возврат, устанавливается параметром Делителя обратной скорости передачи в Пакете инкапсуляции обратной линии передачи данных. Так как сигнал MDDI_Stb переключается со скоростью, равной половине скорости передачи прямой линии передачи данных, тогда скорость передачи обратной линии передачи данных равна половине скорости передачи прямой линии передачи 45 данных, деленной на Делитель обратной скорости передачи. Например, если скорость передачи прямой линии передачи данных составляет 200 Мбит/с и Делитель обратной скорости передачи равен 4, то тогда скорость передачи данных обратной линии передачи данных выражается следующим образом:

$$\frac{1}{2} \cdot \frac{200 \text{ Мбит} / \text{с}}{4} = 25 \text{ Мбит} / \text{с}$$

Пример, показывающий временные соотношения сигнальных линий MDDI_Data0 и MDDI_Stb в Пакете инкапсуляции обратной линии передачи данных, показан на фиг.52, где 50 параметры пакета, используемого для иллюстрации, имеют значения:

Длина пакета = 1024 (0x0400)

Тип пакета = 65 (0x41)

Флаги обратной линии передачи данных = 0

ЦИК параметров = 0xdb43

5 Длина реверсирования передачи 1 = 1

Длина реверсирования передачи 2 = 1

Делитель обратной скорости передачи = 2

Все нули - 0x00

Данные пакета между полями Длины пакета и ЦИК параметров представляют собой:

10 0x00, 0x04, 0x41, 0x00, 0x02, 0x01, 0x01, 0x43, 0xdb, 0x00 ...

Первым пакетом обратной линии передачи данных, возвращаемым от клиента, является Пакет запроса клиента и состояния, имеющий Длину пакета 7, и тип пакета 70. Этот пакет начинается с байтовых значений 0x07, 0x00, 0x46 ... и т.д. На фиг.52, однако, виден только первый байт (0x07). Этот первый пакет обратной линии передачи данных сдвинут во времени почти на один тактовый период обратной линии передачи данных на фигуре для иллюстрации фактической задержки обратной линии передачи данных. Пунктирной линией показана идеальная форма сигнала с нулевой задержкой на двойное прохождение между хостом и клиентом.

Пересылается самый старший байт поля ЦИК параметров, перед которым предшествует 20 тип пакета, затем поле всех нулей. Строб от хоста переключается с единицы на ноль и обратно в единицу, когда данные от хоста меняют уровень, образуя более широкие импульсы. Когда данные переходят в ноль, строб переключается с более высокой скоростью, только изменение в данных на линии данных вызывает изменение около конца поля выравнивания. Строб переключается с более высокой скоростью на остальной части 25 фигуры вследствие фиксированных уровней 0 или 1 сигнала данных в течение расширенных периодов времени, и переходы попадают на структуру импульсной последовательности (фронт).

Тактовый импульс обратной линии передачи данных для хоста находится в нуле до конца периода Реверсирования передачи 1, когда начинаются тактовые импульсы для 30 обеспечения передачи пакетов обратной линии передачи данных. Стрелки на нижней части фигуры обозначают то, когда производится отсчет данных, что будет очевидно из оставшейся части описания. Первый байт пересылаемого поля пакета (в данном случае 11000000) показан начинающимся после поля Реверсирования передачи 1, и уровень на линии стабилизирован от выключенного драйвера хоста. Задержка на прохождение первого 35 бита и, как видно, для бита три может быть видна пунктирными линиями для сигнала Data.

На фиг.53 можно наблюдать типовые значения Делителя обратной скорости передачи, основанные на скорости передачи данных прямой линии передачи данных. Фактический Делитель обратной скорости передачи определяется как результат измерения линии 40 передачи данных на двойное прохождение для гарантирования надлежащей работы обратной линии передачи данных. Первая область 5302 соответствует области безопасной работы, вторая область 5304 соответствует области характеристик в предельных условиях, тогда как третья область 5306 указывает установки, при которых маловероятно, что будет функционировать правильно.

Установки измерения задержки на двойное прохождение и Делителя обратной скорости 45 передачи одинаковые при работе с любой установкой Типа интерфейса как по прямой, так и по обратной линии передачи данных, потому что они выражаются и работают в отношении единиц фактических тактовых периодов, а не количества переданных или принятых битов.

Обычно, максимально возможный Делитель обратной скорости передачи равен 50 половине количества битов, которое может быть послано в окне измерения Пакета измерения задержки на двойное прохождение, используя интерфейс Типа-I, или для данного примера:

$$\frac{(512 \text{ байтов} \cdot 8 \text{ битов/байт})}{2} = 2048$$

Способ улучшенного отсчета обратных данных также может применяться в качестве альтернативы, который позволяет сделать меньшим обратный битовый интервал, чем задержка на двойное прохождение. В отношении этого метода хост не только измеряет задержку на двойное прохождение, но также может определять фазу ответа от клиента относительно «идеальной» границы бита клиента и линии передачи данных с нулевой задержкой. Зная фазу ответа устройства клиента, хост может определить относительно безопасное время для отсчета битов обратных данных от клиента. Измерение задержки на двойное прохождение указывает хосту расположение первого бита обратных данных относительно начала поля Пакетов обратных данных.

На фиг.52B в графической форме иллюстрируется один вариант осуществления примера улучшенного отсчета обратных данных. Идеальный сигнал обратных данных с нулевой задержкой на двойное прохождение показан в виде формы сигнала пунктирной линией. Фактическая задержка на двойное прохождение между 3,5 и 4 циклами MDDI_Stb может наблюдаться как разность в задержке между формой сигнала сплошной линией и идеальной. Это та же самая задержки, которая была бы измерена с использованием Пакета измерения задержки на двойное прохождение, и была бы равной измеренному значению задержки на двойное прохождение, равному 7 битовым интервалам прямой линии передачи данных. В данном варианте осуществления биты обратных данных имеют длину 2 импульсов MDDI_Stb, которые равны 4 битовым интервалам прямой линии передачи данных, что соответствует делителю обратной скорости передачи, равному 2. Для улучшенного отсчета обратных данных обычно используют предварительно выбранный делитель обратной скорости передачи, равный 2, вместо вычисления его, как описано в другом месте. Это, по-видимому, представляет собой по существу оптимальный выбор для улучшенного отсчета обратных данных, так как идеальная точка отсчета легко может определяться с использованием обычных измерений, описанных выше.

Идеальная точка отсчета для обратных данных легко может вычисляться посредством взятия остатка от деления суммарной задержки на двойное прохождение на количество тактовых импульсов прямой линии передачи данных на обратный бит, или задержка на двойное прохождение по модулю тактовых импульсов прямой линии передачи данных на обратный бит. Затем вычесть или 1, или 2 для получения безопасной точки вдали от перехода данных. В данном примере $7 \bmod 4 = 3$, тогда $3-1=2$, или $3-2=1$. Безопасная точка отсчета равна или 1, или 2 битовым интервалам прямой линии передачи данных от фронта «идеальной» границы бита для нулевой задержки на двойное прохождение. Фигура изображает точку отсчета на расстоянии 2 битовых интервалов прямой линии передачи данных от идеальной границы бита, как указано группой вертикальных стрелок в нижней части временной диаграммы. Первая точка отсчета представляет собой первую идеальную границу бита после измеренной задержки на двойное прохождение, плюс смещение для безопасного отсчета. В данном примере измерение задержки на двойное прохождение равно 7, поэтому следующая идеальная граница бита находится на 8-ом битовом интервале, затем добавить или 1, или 2 для безопасной точки отсчета, поэтому отсчет первого бита должен выполняться в или 9, или 10 битовом интервале прямой линии передачи данных после начала Поля Пакетов обратных данных.

XI. Реверсирование передачи и защитные интервалы

Поле Реверсирования передачи 1 в Пакете инкапсуляции обратной линии передачи данных предоставляет время для одновременного выключения драйверов хоста и включения драйверов клиента. Поле Защитного интервала 1 в Пакете измерения задержки на двойное прохождение допускает перекрытие хоста и клиента, поэтому драйверы клиента могут включаться до того, как драйверы интерфейса хоста будут выключены. Поле Реверсирования передачи 2 в Пакете инкапсуляции обратной линии передачи данных дает возможность полной передачи данных в предыдущем поле от клиента, до того как драйверы хоста будут включены. Поле Защитного интервала 2 обеспечивает величину или

период времени, который позволяет драйверам клиента и хоста возбуждаться одновременно уровнем логического нуля. Поля Защитного интервала 1 и Защитного интервала 2, как правило, заполняются предварительно установленными или предварительно выбранными значениями для длин, которые не предназначены для регулировки. В зависимости от используемых аппаратных средств интерфейса эти значения могут быть разработаны, используя эмпирические данные, и откорректированы в некоторых случаях для улучшения работы.

Реверсирование передачи 1

Несколько факторов способствуют определению длины Реверсирования передачи 1, и ими являются скорость передачи данных прямой линии передачи данных и максимальное время выключения драйверов MDDI_Data в хосте, и время включения драйвера клиента, которое, как правило, одинаково, что и время выключения хоста. Длина поля Реверсирования передачи 1 выбирается равной $24 \cdot t_{\text{BIT}}$ (таблица XI). Длина количества байтов прямой линии передачи данных поля Реверсирования передачи 1 определяется с использованием Множителя типа интерфейса и вычисляется с использованием зависимости:

$$\text{Длительность}_{\text{Реверсирование Передачи 1}} = \frac{24}{8 \text{ бит/байт}} \cdot \text{Множитель Типа Интерфейса}_{\text{ПРМ}} = 3 \cdot \text{Множитель Типа Интерфейса}_{\text{ПРМ}}$$

где Множитель типа интерфейса равен 1 для Типа 1, 2 для Типа 2, 4 для Типа 3 и 8 для Типа-4.

Реверсирование передачи 2

Факторами, которые определяют продолжительность времени, как правило, используемую для Реверсирования передачи 2, являются задержка на двойное прохождение линии обмена данными, максимальное время выключения драйверов MDDI_Data в клиенте и время включения драйвера хоста, который задается равным времени выключения драйвера клиента. Максимальное время включения драйвера хоста и время выключения драйвера клиента задается в другом месте. Задержка на двойное прохождение измеряется в единицах t_{BIT} . Минимальная длина, задаваемая количеством байтов прямой линии передачи данных поля Реверсирования передачи 2, вычисляется согласно зависимости:

$$\text{Длительность}_{\text{Реверсирование Передачи 2}} \geq \text{Округлить До Следующего Целого Числа} \left(\frac{\text{Задержка На Двойное Прохождение} + 24}{8 \text{ битов/байт}} \cdot \text{Множитель Типа Интерфейса}_{\text{ПРМ}} \right)$$

Например, прямая линия передачи данных Типа 3 с задержкой на двойное прохождение 10 тактовых импульсов прямой линии передачи данных использует задержку Реверсирования передачи 2 порядка:

$$\text{Длительность}_{\text{Реверсирование Передачи 2}} \geq \text{Округлить До Следующего Целого Числа}$$

$$\left(\frac{11 + 24}{8} \cdot 4 \right) = 18 \text{ байтов}$$

XII. Альтернативные временные соотношения обратной линии передачи данных

Хотя использование временных соотношений и защитных полос, описанных выше, работает для достижения интерфейса с высокой скоростью пересылки данных, изобретатели обнаружили метод для учета длительности обратных битов, которые короче, чем время двойного прохождения, посредством изменения обнаружения обратных временных соотношений.

Как представлено выше, предыдущий подход к временным соотношениям обратной линии передачи данных конфигурируется так, что подсчитывается количество тактовых циклов с последнего бита Защитного интервала 1 обратного пакета временных соотношений до тех пор, пока не будет выполнен отсчет первого бита на переднем фронте тактового импульса ввода-вывода. Т.е. тактовый сигнал(ы), используемый для согласования во времени входов и выходов для ЦИМД. Вычисление делителя обратной

скорости передачи тогда определяется следующим образом:

$$\text{делитель_обратной_скорости_передачи} = \text{ОкруглениеДоследующегоЦелогочисла} = \left(\frac{\text{задержка_на_двойное_прохождение} + 1}{2} \right)$$

5 Это обеспечивает ширину бита, равную задержке на двойное прохождение, которая приводит к очень надежной обратной линии передачи данных. Однако, как было показано, обратная линия передачи данных способна работать быстрее или при более высокой скорости передачи данных, чем изобретатели и хотят воспользоваться. Новый метод изобретения позволяет использовать дополнительные возможности интерфейса для

10 достижения более высокого быстродействия. Это достигается тем, что хост подсчитывает количество тактовых циклов до тех пор, пока выполняется отсчет единицы, но хост выполняет отсчет линии данных как на переднем фронте, так и на заднем фронте во время обратного пакета временных соотношений. Это дает возможность хосту выбрать наиболее полезную или даже

15 оптимальную точку отсчета в пределах обратного бита для обеспечения того, что бит является стабильным. Т.е. найти наиболее полезный или оптимальный передний фронт для выполнения отсчета данных для пакетов инкапсуляции обратной линии передачи данных обратного трафика. Оптимальная точка выполнения отсчета зависит как от делителя обратной линии передачи данных, так и от того, была ли обнаружена первая

20 единица на переднем фронте или на заднем фронте. Новый способ согласования во времени позволяет хосту просто искать первый фронт комбинации 0xFF 0xFF 0x00, посылаемой клиентом для согласования во времени обратной линии передачи данных для определения, где выполнять отсчет в пакете инкапсуляции обратной линии передачи данных.

25 Примеры поступающего обратного бита и как этот бит будет выполнять поиск различных делителей обратной скорости передачи, изображены на фиг.64 вместе с количеством тактовых циклов, которые произошли с момента последнего бита Защитного интервала 1. На фиг.64 можно видеть, что, если первый фронт происходит между передним и задним фронтом (обозначенными как передний/задний), оптимальной точкой отсчета для делителя

30 обратной скорости передачи, равного единице, оптимальная точка отсчета представляет собой фронт тактового цикла, обозначенный «b», так как он представляет собой единственный передний фронт, происходящий в течение периода обратного бита. Для делителя обратной скорости передачи, равного двум, оптимальной точкой отсчета, вероятно, является все же передний фронт «b» тактового цикла, так как фронт «с» цикла

35 ближе к фронту бита, чем «b». Для делителя обратной скорости передачи, равного четырем, оптимальной точкой отсчета, вероятно, является фронт «d» тактового цикла, так как он ближе к заднему фронту обратного бита, где значение, вероятно, стабилизировалось.

40 Возвращаясь к фиг.64, если, однако, первый фронт происходит между задним и передним фронтом (обозначенными как задний/передний), оптимальной точкой отсчета для делителя обратной скорости передачи, равного единице, является фронт «a» тактового цикла точки отсчета, как тот, который является единственным передним фронтом в пределах периода обратного битового интервала. Для делителя обратной скорости передачи, равного двум, оптимальной точкой отсчета является фронт «b», и для делителя

45 обратной скорости передачи, равного четырем, оптимальной точкой отсчета является фронт «с».

Можно видеть, что, по мере того как делители обратной скорости передачи становятся все больше и больше, оптимальную точку отсчета становится легче обнаружить или

50 выбрать, так как она должна быть передним фронтом, который является самым ближайшим к середине.

Хост может использовать этот метод для нахождения количества передних фронтов тактовых импульсов, перед тем как будет наблюдаться передний фронт данных у данных пакета согласования во времени на линии данных. Затем можно принять решение,

основываясь на том, происходит ли фронт между передним и задним фронтом или между задним и передним фронтом, и чему равен делитель обратной скорости передачи, сколько дополнительных тактовых циклов добавить к счетчику количества, чтобы целесообразно обеспечить, чтобы отсчет бита всегда производился максимально возможно близко к

5 середине.

Если хост выбрал или определил количество тактовых циклов, он может «исследовать» различные делители обратной скорости передачи с клиентом для определения, будет ли работать конкретный делитель обратной скорости передачи. Хост (и клиент) может начать с делителя, равного единице, и проверить ЦИК обратного пакета состояния, принятого от
10 клиента, для определения, правильно ли функционирует эта обратная скорость передачи для пересылки данных. Если ЦИК поврежден, существует, вероятно, ошибка отсчета, и хост может увеличить делитель обратной скорости передачи и попытаться снова запросить пакет состояния. Если второй запрошенный пакет поврежден, делитель снова может быть увеличен и снова выполнен запрос. Если этот пакет декодируется правильно, этот
15 делитель обратной скорости передачи может использоваться для всех будущих обратных пакетов.

Этот способ является эффективным и полезным, так как обратное согласование во времени не должно меняться с начальной оценки согласования во времени по двойному прохождению. Если прямая линия передачи данных стабильная, клиент должен
20 продолжать декодирование пакетов прямой линии передачи данных, даже если имеются сбои обратной линии передачи данных. Конечно, все же является ответственностью хоста установка делителя обратной линии передачи данных для линии передачи данных, так как этот способ не гарантирует совершенную обратную линию передачи данных. Кроме того, делитель зависит, главным образом, от качества генератора тактовых импульсов, который
25 используется для генерирования тактовых импульсов ввода-вывода. Если этот генератор тактовых импульсов имеет значительную величину дрожания, то существует большая возможность ошибки отсчета. Эта возможность ошибки увеличивается с количеством тактовых циклов в задержке на двойное прохождение.

Эта реализация, по-видимому, наилучшим образом работает для обратных данных Типа
30 1, но может представлять проблемы для обратных данных Типа 2-Типа 4 вследствие перекаса между линиями данных, потенциально слишком большими для работы линии передачи данных со скоростью передачи, которая работает наилучшим образом только для одной пары данных. Однако нет необходимости, вероятно, снижать скорость передачи данных до предыдущего способа даже с Типом 2-Типом 4 для работы. Этот способ также
35 может работать наилучшим образом, если дублируется на каждой линии данных для выбора идеального или оптимального расположения отсчета тактового импульса. Если они происходят в один и тот же момент времени отсчета для каждой пары данных, этот способ будет продолжать работать. Если они происходят в различные периоды отсчета, могут использоваться два различных подхода. Первым является выбор требуемого или более
40 оптимизированного расположения отсчета для каждой точки данных, даже если она не одинакова для каждой пары данных. Хост затем может восстановить поток данных после выполнения отсчета всех битов из набора пар данных: два бита для Типа 2, четыре бита для Типа 3 и восемь битов для Типа 4. Другой вариант выбора предназначен для хоста и заключается в увеличении делителя обратной скорости передачи данных, так что может
45 выполняться отсчет битов данных для каждой пары данных на одном и том же фронте тактового импульса.

XIII. Влияние задержки и перекаса линии передачи данных

Перекас задержки на прямой линии передачи данных между парами MDDI_Data и MDDI_Stb может ограничить максимально возможную скорость передачи данных, если не
50 используется компенсация перекаса задержки. Различия в задержке, которые вызывают перекас временных соотношений, происходят вследствие логики контроллера, линейных драйверов и приемников и кабеля и соединителей, как вкратце изложено ниже.

A. Анализ временных соотношений линии передачи данных, ограниченный перекасом

(Тип-1 ЦИМД)

1. Пример задержки и перекося линии передачи данных Типа 1

Типовая схема интерфейса, подобная той, которая показана на фиг.41, изображена на фиг.57 для обеспечения линии передачи данных интерфейса Типа 1. На фиг.57 примерные или типовые значения для задержки на распространение и перекося показаны для каждого из нескольких этапов обработки или интерфейса прямой линии передачи данных ЦИМД Типа 1. Перекос в задержке между MDDI_Stb и MDDI_Data0 вызывает искажение рабочего цикла выходных тактовых импульсов. Данные на D-входе триггерного каскада (RXFF) приемника, использующего триггеры 5728, 5732, изменяются незначительно после фронта тактового импульса, так что может надежно выполняться их отсчет. Фигура изображает две каскадно включенные линии 5732a и 5732b задержки, используемые для решения двух различных проблем при помощи создания этой времязадающей зависимости. При фактической реализации они могут объединяться в один элемент задержки.

На фиг.58 изображены Data, Stb и Временные соотношения восстановления Clock (тактовых импульсов) на Линии передачи данных Типа 1 для примерной обработки сигнала при помощи интерфейса.

Суммарный перекося задержки, который является значительным, как правило, возникает или происходит из суммы перекося в следующих каскадах: триггеров (TXFF) передатчика с триггерами 5704, 5706; драйверов (TXDRV) передатчика с драйверами 5708, 5710; Кабеля 5702; линейных приемников (RXRCVR) приемника с приемниками 5722, 5724; и логики (RXOR) исключяющее ИЛИ приемника. Задержка 5732a должна соответствовать или превышать задержку вентиля 5736 исключяющего ИЛИ в каскаде RXOR, которая определяется зависимостью:

$$t_{PD-\min(Delay\ 1)} \geq t_{PD-\max(XOR)}$$

Желательно выполнить требование, чтобы D-вход триггеров 5728, 5732 приемника не изменялся до его входа тактовых импульсов. Это действительно, если время удержания RXFF равно нулю.

Назначением или функцией Задержки 2 является компенсация времени удержания триггера RXFF согласно зависимости:

$$t_{PD-\min(Delay\ 2)} = t_{H(RXFF)}$$

Во многих системах она равна нулю, так как время удержания равно нулю, и, конечно, в этом случае максимальная задержка Задержки 2 также может быть равна нулю.

Вклад наихудшего случая в перекося в каскаде XOR приемника равен случаю поздних данных/раннего строба, когда Задержка 1 имеет максимальное значение, и тактовый выходной сигнал с вентиля исключяющее ИЛИ (XOR) поступает максимально возможно рано согласно зависимости:

$$t_{SKEW-\max(RXOR)} = t_{PD-\max(Delay\ 1)} - t_{PD-\min(XOR)}$$

В данном случае данные могут измениться между двумя битовыми периодами, n и n+1, очень близко к моменту времени, где бит n+1 тактируется в триггер приемника.

Максимальная скорость передачи данных (минимальный битовый период) линии передачи данных ЦИМД Типа 1 является функцией максимального перекося, встречаемого по всем драйверам, кабелям и приемникам в линии передачи данных ЦИМД плюс суммарная установка данных в каскаде RXFF. Суммарный перекося задержки в линии передачи данных до выхода каскада RXRCVR может быть выражен следующим образом:

$$t_{SKEW-\max(LINK)} = t_{SKEW-\max(TXFF)} + t_{SKEW-\max(TXDRV)} + t_{SKEW-\max(CABLE)} + t_{SKEW-\max(RXRCVR)}$$

причем «кабель» представляет многочисленные проводники, или межсоединения, или

провода и соответствующую задержку, и минимальный битовый период определяется следующим образом:

$$t_{BIT-min} = t_{SKEW-max(LINK)} + 2 \cdot t_{B-TP4} + t_{Asymmetry} + t_{SKEW-max(RX XOR)} + t_{jitter-host} + t_{PD-max(Delay2)} + t_{SU(RXFF)}$$

В примере, показанном на фиг.57 для внешнего режима, $t_{SKEW-max(LINK)}=1000$ пс и минимальный битовый период может быть выражен следующим образом:

$$t_{BIT-min} = 1000 + 2 \cdot 125 + 625 + 125 + 200 + 0 + 100 = 2300 \text{ пс}$$

или указано как примерно 434 Мбит/с. В примере, показанном на фиг.57 для внутреннего режима, $t_{SKEW-max(LINK)}=500$ пс и минимальный битовый период может быть выражен следующим образом:

$$t_{BIT-min} = 500 + 2 \cdot 125 + 625 + 125 + 200 + 0 + 100 = 1800 \text{ пс}$$

или указано как примерно 555 Мбит/с.

В. Анализ временных соотношений линии передачи данных для Типа 2, 3 и 4 ЦИМД

Типовая схема интерфейса, подобная той, которая показана на фиг.41 и 57, изображена на фиг.59 для обеспечения линий передачи данных интерфейса Типа 2, 3 и 4.

Дополнительные элементы используются в каскадах TXFF (5904), TXDRVR (5908), RXRCVCR (5922) и RXFF (5932, 5928, 5930) для обеспечения дополнительной обработки сигнала. На фиг.59 показаны примерные или типовые значения для задержки на распространение и перекося для каждого из нескольких каскадов обработки или интерфейса прямой линии передачи данных ЦИМД Типа 2. В дополнение к перекося задержки между MDDI_Stb и MDDI_Data0, оказывающей влияние на рабочий цикл выходных тактовых импульсов, также существует перекося между обоими из этих двух сигналов и другими сигналами MDDI_Data. Данные на D-входе триггерного В (RXFFB) каскада приемника, состоящего из триггеров 5928 и 5930, изменяются незначительно после фронта тактового импульса, поэтому отсчет их может выполняться надежно. Если MDDI_Data1 поступает ранее MDDI_Stb или MDDI_Data0, тогда MDDI_Data1 должен быть задержан для выполнения его отсчета по меньшей мере на величину перекося задержки. Чтобы это выполнить, данные задерживаются с использованием линии задержки Задержка 3. Если MDDI_Data1 поступает позже MDDI_Stb и MDDI_Data0 и он также задерживается на Задержку 3, тогда точка, где MDDI_Data1 изменяется, перемещается ближе к фронту следующего тактового импульса. Этот процесс определяет верхний предел скорости передачи данных линии передачи данных ЦИМД Типа 2, 3 или 4. Некоторые примерные различные возможности для зависимостей согласования во времени и перекося двух сигналов данных и MDDI_Stb относительно друг друга изображены на фиг.60A, 60B и 60C.

Чтобы надежно выполнить отсчет данных в RXFFB, когда MDDI_DataX поступает максимально возможно рано, Задержка 3 устанавливается согласно зависимости:

$$t_{PD-min(Delay3)} \geq t_{SKEW-max(LINK)} + t_{H(RXFFB)} + t_{PD-max(XOR)}$$

Максимальное быстродействие линии передачи данных определяется минимально допустимым битовым периодом. На это больше всего оказывается влияние, когда MDDI_DataX поступает насколько возможно поздно. В этом случае минимально допустимая длительность цикла определяется следующим образом:

$$t_{BIT-min} = t_{SKEW-max(LINK)} + t_{PD-max(Delay3)} + t_{SU(RXFFB)} - t_{PD-min(XOR)}$$

Верхняя граница быстродействия линии передачи данных тогда равна:

$$t_{PD-max(Delay3)} = t_{PD-min(Delay3)}$$

и при наличии этого предположения:

$$t_{BIT-\min(lower-bound)} = 2 \cdot t_{SKEW-\max(LINK)} + t_{PD-\max(XOR)} + t_{SU(RXFFB)} + t_{H(RXFFB)}$$

- 5 В приведенном выше примере нижняя граница минимального битового периода определяется зависимостью:

$$t_{BIT-\min(lower-bound)} = 2 \cdot (1000 + 2 \cdot 125 + 625 + 200) + 1500 + 100 + 0 = 5750 \text{ пс} ,$$

- 10 что примерно равно 174 Мбит/с.

- 10 Это значительно медленнее, чем максимальная скорость передачи данных, которая может использоваться с линией передачи данных Типа 1. Возможность автоматической компенсации перекося задержки в ЦИМД существенно снижает воздействие, которое перекося задержки оказывает на фактор максимальной скорости передачи линии передачи данных, который как раз на границе установки действительных данных. Калиброванный перекося между MDDI_Data0 и MDDI_Stb равен:

$$t_{SKEW-\max(Calibrated)} = 2 \cdot t_{TAP-SPACING-\max} ,$$

и минимальный битовый период равен:

$$20 \quad t_{BIT-\min-Calibrated} = t_{SKEW-\max(Calibrated)} + 2 \cdot t_{B-TP4} + t_{Asymmetry} + t_{jitter-host} + t_{SKEW-\max(RXAND+KXXOR)} + t_{SU(RXFF)}$$

- Где «ТВ» или t_B представляет дрожание сигнала от границы бита до минимального выходного уровня. Асимметрия просто ссылается на асимметричную сущность внутренней задержки дифференциальных приемников или вследствие их. «ТР4» ассоциируется или 25 эффективно определяется для целей снятия и тестирования электрических характеристик в качестве соединения или интерфейса (выводы устройства контроллера ЦИМД в клиенте) для дифференциальных линейных драйверов и приемников для клиента. Он представляет удобную или предварительно определенную точку, от которой задержка сигнала 30 измеряется и составляются характеристики для линии передачи данных по всей остальной части системы. В одном варианте осуществления максимальное значение параметра t_B в TP4 определяется зависимостью

$$t_{Differential-Skew-TP4-DRVR-EXT} = 0,3 \cdot t_{BIT} \text{ для внешнего режима и}$$

$$t_{Differential-Skew-TP4-DRVR-INT} = 0,6 \cdot t_{BIT} \text{ для внутреннего режима для передатчиков клиента; и}$$

$$35 \quad t_{B-TP4-RCVR-EXT} = 0,051 \cdot t_{BIT} + 175 \text{ пс для внешнего режима для приемников клиента.}$$

- Метка TP4 просто является полезной при перечислении различных контрольных точек (TP) в интерфейсе и линиях передачи данных. В одном варианте осуществления 40 расположение этой контрольной точки определяется одинаковым как для внешнего, так и для внутреннего режимов. Существует соответствующая контрольная точка «TP0» для подсоединения выводов интерфейса устройства контроллера ЦИМД в хосте или связанная с ними, которая содержит дифференциальные линейные драйверы и приемники. В данном варианте осуществления максимальное значение параметра T_B в TP0 определяется 45 зависимостью $t_{B-TP0-RCVR-INT} = 0,051 \cdot t_{BIT} + 50$ пс для внутреннего режима, и $t_{B-TP0-RCVR-EXT} = 0,051 \cdot t_{BIT} + 175$ пс для внешнего режима для приемников хоста; и $t_{B-TP0} = 0,102 \cdot t_{BIT}$ для передатчиков хоста.

В примере, показанном на фиг.59, $t_{SKEW-\max(Data-Stb-Calibrated)} = 300$ пс и минимальный битовый период:

$$t_{BIT-\min-Calibrated} = 300 + 2 \cdot 125 + 625 + 200 + 175 + 100 = 1650 \text{ пс} ,$$

- 50 примерно 606 Мбит/с.

Чтобы надежно выполнять отсчет данных в RXFFB, когда MDDI_Data1 поступает максимально рано, связанная с ним программируемая задержка регулируется на

оптимальную установку с точностью одного отвода и для безопасности добавляется задержка дополнительного отвода. Максимальное быстродействие линии передачи данных определяется минимально допустимым битовым периодом. Это оказывает наибольшее влияние, когда MDDI_Data1 поступает максимально поздно. В этом случае минимально допустимая длительность цикла равна:

$$t_{BIT-min-Data1-Calibrated} = 2 \cdot t_{TAP-Spacing-max} + 2 \cdot t_{TA-TP4},$$

где «ТА» или t_A представляет дрожание сигнала от границы бита до центрального пересечения.

В примере, приведенном на фиг.59, нижняя граница минимального битового периода, основываясь на выполнении отсчета MDDI_Data1, равна:

$$t_{BIT-min-Data1-Calibrated} = 2 \cdot 150 + 2 \cdot 125 = 550 \text{ пс}$$

В одном варианте осуществления типовое суммарное время задержки для перекоса задержки, асимметрии задержки и Дрожания тактовых импульсов в передатчике хоста для Внутреннего режима определяется следующим образом:

$$t_{Asymmetry-TXFF} + t_{Asymmetry-TXDRVR} + t_{Skew-TXFF} + t_{Skew-TXDRVR} + t_{jitter-host} = 0.467 \cdot (t_{BIT} - 150 \text{ пс}),$$

и для внешнего режима:

$$t_{Asymmetry-TXFF} + t_{Asymmetry-TXDRVR} + t_{Skew-TXFF} + t_{Skew-TXDRVR} + t_{jitter-host} = 0.7BD \cdot (t_{BIT} - 150.7BD \text{ пс})$$

тогда как типовое суммарное время задержки для перекоса задержки, асимметрии задержки и времени установления в устройстве клиента (t_{B-TP4}) для внутреннего режима равно:

$$t_{Asymmetry-RXRCVR} + t_{Asymmetry-RXXOR} + t_{Skew-RXRCVR} + t_{Skew-RXXOR} + t_{setup-RXFF} = 0.307 \cdot (t_{BIT} - 150 \text{ пс})$$

и для внешнего режима:

$$t_{Asymmetry-RXRCVR} + t_{Asymmetry-RXXOR} + t_{Skew-RXRCVR} + t_{Skew-RXXOR} + t_{setup-RXFF} = 0.7BD \cdot (t_{BIT} - 7BD \text{ пс})$$

где член TBD представляет собой гибкую метку сохранения места для подлежащих определению в будущем значений, которые будут зависеть от многочисленных хорошо понятных характеристик и эксплуатационных требований для подключений внешнего режима.

XIV. Описание межсоединений физического уровня

Физические соединения, полезные для выполнения интерфейса в соответствии с настоящим изобретением, могут быть реализованы, используя серийно выпускаемые элементы, такие как элемент с номером 3260-8S2(01), производимый компанией Hirose Electric Company LTD, на стороне хоста и элемент с номером 3240-8P-C, производимый компанией Hirose Electric Company LTD, на стороне устройства клиента. Примерное назначение выводов или «цоколевка» интерфейса для таких соединителей, используемых с интерфейсами Тип-1/Тип 2, перечислено в таблице XIII и изображено на фиг.61.

Таблица XIII			
Название сигнала	Номер вывода	Название сигнала	Номер вывода
MDDI_Pwr	1	MDDI_Gnd	11
MDDI_Stb+	2	MDDI_Stb-	12
MDDI_DATA0+	4	MDDI_DATA0-	14
MDDI_DATA1+	6	MDDI_DATA1-	16
MDDI_DATA2+	8	MDDI_DATA2-	18
MDDI_DATA3+	10	MDDI_DATA3-	20
MDDI_DATA4+	9	MDDI_DATA4-	19
MDDI_DATA5+	7	MDDI_DATA5-	17

MDDI_DATA6+	5	MDDI_DATA6-	15
MDDI_DATA7+	3	MDDI_DATA7-	13
		Экран	

Экран подсоединен к HOST_Gnd в интерфейсе хоста, и дренирующий провод экрана в кабеле подсоединен к экрану соединителя клиента. Однако экран и дренирующий провод не подсоединены к схемному заземлению внутри клиента.

Элементы или устройства межсоединений выбраны или разработаны такими, чтобы они были достаточно маленькими для использования с мобильными устройствами обмена данными и вычислительными устройствами, такими как ПЦП и беспроводные телефоны, или портативные игровые устройства, при этом чтобы не были вызывающими и неэстетичными по сравнению с относительным размером устройства. Любые соединители и кабели должны быть достаточно прочными для использования в типовых условиях эксплуатации у потребителя и допускали малый размер, особенно для кабеля, и относительно низкую стоимость. Элементы передачи должны обеспечивать сигналы данных и строб-сигналы, которые представляют собой дифференциальные данные БВН, имеющие скорость передачи до примерно 450 Мбит/с для Типа 1 и Типа 2 и до 3,6 Гбит/с для 8-битового параллельного варианта Типа 4.

Для применений внутреннего режима нет ни соединителей в этом смысле для используемых проводников, ни таких соединительных элементов, которые имеют тенденцию быть очень миниатюрными. Одним примером является «гнезда» с нулевым усилием сочленения и принудительным обжатием для установки интегральных схем или элементов, в которые вставляется или устройство хоста, или устройство клиента. Другим примером является то, когда хост и клиент находятся на печатных платах с различными проводниками межсоединений и имеют «выводы» или контакты, выступающие из корпусов, которые припаиваются к контактам на проводниках для внутренних соединений интегральных схем.

XV. Принцип действия

На фиг.54А и 54В показано краткое изложение общих этапов, предпринимаемых при обработке данных и пакетов во время работы интерфейса, используя варианты осуществления изобретения, вместе с обзором устройства интерфейса, обрабатывающего пакеты, на фиг.55. На этих фигурах процесс начинается на этапе 5402 с определения, соединены ли или нет клиент и хост с использованием тракта обмена данными, в данном случае кабеля. Это может происходить посредством использования периодического опроса хостом, используя программное обеспечение или аппаратные средства, которые обнаруживают присутствие соединителей или кабелей, или сигналов на входах хоста (как, например, с интерфейсами УПШ), или другими известными методами. Если клиент не подсоединен к хосту, то тогда он просто может перейти в состояние ожидания некоторой предварительно определенной длительности в зависимости от применения, перейти в режим «спячки» или быть неактивизированным и ожидать будущего использования, которое может потребовать от пользователя предпринять некоторое действие для повторного активизирования хоста. Например, когда хост постоянно находится на устройстве компьютерного типа, то пользователю, может быть, необходимо будет нажать кнопкой мыши на пиктограмме экрана или запросить программу, которая активизирует обработку хоста по поиску клиента. Также простая установка в соединение типа УПШ может активизировать обработку хостом в зависимости от возможностей и конфигурации хоста или резидентного программного обеспечения хоста.

Если клиент подсоединен к хосту или, наоборот, обнаружен как присутствующий, то или клиент, или хост посылает соответствующие пакеты, запрашивающие обслуживание на этапах 5404 и 5406. Клиент может послать или пакеты Запроса на обслуживание или состояния клиента на этапе 5404. Отмечается, что линия передачи данных, как описано ранее, ранее могла быть закрыта или может находиться в режиме «спячки», так что это может быть не полной инициализацией линии обмена данными, которая последует. Если

линия обмена данными синхронизирована и хост пытается установить обмен данными с клиентом, клиент также предоставляет хосту пакет Возможностей клиента, как на этапе 5408. Хост теперь может начать определение типа поддержки, включая скорости пересылки, которые клиент может обеспечить.

5 В общих чертах на этапе 5410 хост и клиент также согласовывают тип (скорости передачи/быстродействие) используемого режима обслуживания, например, Тип 1, Тип 2 и т.д. Если тип обслуживания установлен, хост может начинать пересылку информации. Кроме того, хост может использовать Пакеты измерения задержки на двойное прохождение для оптимизации временных соотношений линий обмена данными параллельно с другой
10 обработкой сигнала, как показано на этапе 5411.

Как указано ранее, все пересылки начинаются с Пакета заголовка подкадра, показанного передаваемым на этапе 5412, за которым следует тип данных, в данном случае пакеты видео- и аудиопотока, и пакеты заполнителя, показанные передаваемыми на этапе 5414. Аудио- и видеоданные предварительно были подготовлены или отображены
15 в пакеты, и пакеты заполнителя вводятся по необходимости или по требованию для заполнения требуемого количества битов для медиакадров. Хост может посылать пакеты, такие как Пакеты включения прямых аудиоканалов, для активизации звуковых устройств. Кроме того, хост может пересылать команды и информацию, используя описанные выше другие типы пакетов, в данном случае показанные как пересылка пакетов Карты цветов,
20 Пересылки битовых блоков или других пакетов на этапе 5416. Кроме того, хост и клиент могут обмениваться данными, относящимися к клавиатуре или указательным устройствам, используя соответствующие пакеты.

Во время работы может произойти одно из нескольких различных событий, которые ведут к тому, что хосту или клиенту требуются различные скорости передачи данных или
25 тип режима интерфейса. Например, компьютер или другое устройство, выполняющее обмен данными, могут встретить условия загрузки при обработке данных, которые вызывают замедление в подготовке или представлении пакетов. Устройство клиента, принимающее данные, может переключиться со специализированного источника питания от сети переменного тока на более ограниченный источник питания от батарей и или не
30 сможет передавать данные так быстро, обрабатывать команды так быстро, или не сможет использовать тот же уровень разрешения или глубину цвета при более ограниченных установках питания. Альтернативно ограничивающее условие может уменьшиться или исчезнуть, позволяя любому устройству пересылать данные при более высоких скоростях передачи. Поскольку это более желательно, то запрос может быть сделан на переключение
35 в режим с более высокой скоростью передачи.

Если имеют место или изменяются эти или другие типы известных условий, то или хост, или клиент могут их обнаружить и попытаться повторно согласовать режим интерфейса. Это показано на этапе 5420, где хост посылает Пакеты запроса перехода на тип
40 интерфейса клиенту, запрашивая переход в другой режим, клиент посылает Пакеты подтверждение типа интерфейса, подтверждающие, что изменение выполняется и хост посылает Пакеты выполнения перехода на тип для выполнения переключения в заданный режим.

Несмотря на то что не требуется конкретный порядок обработки, клиент и хост также могут обмениваться пакетами, относящимися к данным, предназначенным для или
45 принимаемым от указательных устройств, клавиатур и других устройств ввода пользовательского типа, связанных, главным образом, с клиентом, хотя такие элементы также могут присутствовать и на стороне хоста. Эти пакеты обычно обрабатываются с использованием элемента типа общего процессора, а не конечного автомата (5502). Кроме того, некоторые из описанных выше команд также будут обрабатываться общим
50 процессором (5504, 5508).

После того как будет произведен обмен данными и командами между хостом и клиентом, в некоторый момент принимается решение, должны ли или нет дополнительные данные передаваться, или хост или клиент собираются прекратить обслуживание

пересылки. Это показано на этапе 5422. Если линия собирается перейти в состояние «спячки» или быть полностью выключенной, хост посылает пакет Закрытия линии передачи данных клиенту и обе стороны завершают пересылку данных.

Пакеты, пересылаемые при обработке вышеупомянутыми операциями, будут
 5 пересылаться с использованием драйверов и приемников, ранее описанных в отношении контроллеров хоста и клиента. Эти линейные драйверы и другие логические элементы подключаются к конечному автомату и общим процессорам, описанным выше, как изображено на общем виде на фиг.55. На фиг.55 конечный автомат 5502 и общие
 10 процессоры 5504 и 5508 дополнительно могут быть подсоединены к другим непоказанным элементам, таким как специализированный интерфейс УПШ, элементы памяти или другие компоненты, постоянно находящиеся вне контроллера линии передачи данных, с которыми они взаимодействуют, включая, но не ограничиваясь ими, источник данных и кристаллы управления видео для устройств просмотра на дисплее.

Процессоры и конечный автомат обеспечивают управление включением и выключением
 15 драйверов, как описано выше в отношении защитных интервалов и т.д., для обеспечения эффективного установления или завершения линии обмена данными и пересылки пакетов.

XVI. Буферы кадров дисплея

Требования на буферизацию видеоданных различны для движущихся
 видеоизображений по сравнению с компьютерной графикой. Данные пикселей наиболее
 20 часто хранятся в локальном буфере кадров в клиенте, поэтому изображения на клиенте могут обновляться локально.

Когда видео с полноценным движением отображается (почти каждый пиксель в дисплее
 изменяется с каждым Медиакадром), обычно предпочтительно хранить поступающие
 данные пикселей в одном буфере кадра, тогда как изображение на дисплее обновляется из
 25 второго буфера кадра. Более двух буферов дисплея могут использоваться для устранения видимых артефактов, как описано ниже. Когда все изображение будет принято в один буфер кадра, тогда роли буферов могут меняться местами, и вновь принятое изображение тогда используется для обновления изображения дисплея, и другой буфер заполняется следующим кадром изображения. Этот принцип иллюстрируется на фиг.88А, где данные
 30 пикселей записываются в автономный буфер изображения посредством установки битов Обновления изображения дисплея в «01».

В других применениях хосту необходимо обновить только небольшую часть
 изображения без необходимости перерисовки всего изображения. В данном случае
 35 желательно записывать новые пиксели непосредственно в буфер, используемый для обновления изображения дисплея, как подробно изображено на фиг.88В.

В применениях, которые имеют фиксированное изображение с небольшим видеоокном,
 проще всего записать фиксированное изображение в оба буфера (биты обновления
 изображения дисплея равны «11»), как показано на фиг.88С, и впоследствии записывать
 40 пиксели движущегося изображения в автономный буфер посредством установки битов обновления изображения дисплея в «01».

Следующие правила описывают полезное манипулирование указателями буферов,
 одновременно записывая новую информацию клиенту и обновляя изображение дисплея.
 Существует три указателя буферов: `current_fill` указывает на буфер, заполняемый в
 настоящий момент данными по линии передачи данных ЦИМД. `Just_filled` указывает на
 45 буфер, который был заполнен самым последним, `being_displayed` указывает на буфер, используемый в настоящий момент для обновления изображения дисплея. Все три указателя буфера могут содержать значения от 0 до N-1, где N представляет собой число буферов дисплея, и $N \geq 2$. Арифметическим действием над указателями буферов является `mod N`, например, когда $N=3$ и `current_fill=2`, приращение `current_fill` вызывает
 50 установку `current_fill` в 0. В простом случае, когда $N=2$, `just_filled` всегда представляет собой дополнительный код для `current_fill`. При каждой границе Медиакадра ЦИМД (Пакет заголовка подкадра с полем Число подкадров, равным поэтому нулю) выполнить следующие операции в заданном порядке: установить `just_filled` равным

current_fill и установить current_fill равным current_fill+1.

Пакеты видеопотока ЦИМД обновляют буферы согласно структуре или методологии: когда Биты обновления изображения дисплея равны «01», данные пикселей записываются в буфер, задаваемый посредством current_fill; когда Биты обновления изображения

5 дисплея равны «00», данные пикселей записываются в буфер, задаваемый посредством just_filled; и когда Биты обновления изображения дисплея равны «11», данные пикселей записываются во все буферы. Изображение дисплея обновляется из буфера, задаваемого указателем being_displayed. После того как дисплей обновит последний пиксель в эпохе обновления одного кадра и перед тем, как он начнет обновлять первый пиксель в эпохе
10 обновления следующего кадра, процесс обновления изображения дисплея выполняет операцию установки being_refreshed равным just_filled.

Пакеты с полем Атрибута данных пикселей содержат пару Битов обновления изображения дисплея, которые задают буфер кадра, где данные пикселей должны быть записаны. Пакет возможностей клиента имеет три дополнительных бита, которые
15 указывают, какие комбинации Битов обновления изображения дисплея поддерживаются в клиенте. Во многих случаях генерируемые компьютером изображения требуют инкрементного обновления, основанного на вводе пользователем или извлекаемого из информации, принятой от компьютерной сети. Комбинации «00» и «11» Битов обновления изображения дисплея поддерживают этот режим работы, вызывая запись данных пикселей
20 в отображаемый буфер кадра или в оба буфера кадра.

При обеспечении видеоизображений фиг.89 иллюстрирует то, как видеоизображения отображаются с использованием пары буферов кадра, когда видеоданные передаются по линии передачи данных ЦИМД с Битами обновления изображения дисплея, равными «01». После того как будет обнаружена граница медиакадра на линии передачи данных ЦИМД,
25 процесс обновления изображения дисплея начинает обновление из буфера следующего кадра, когда завершен процесс обновления для обновляемого в настоящий момент кадра.

Важным предположением, относящимся к фиг.89, является то, что изображение принимается от хоста в виде непрерывного потока пикселей, которые передаются в том же порядке, в каком клиент использует для считывания пикселей из буфера кадра для
30 обновления изображения дисплея (обычно с верхнего левого угла, считывая построчно до нижнего правого угла экрана). Это представляет собой важную деталь в тех случаях, где операции Обновления изображения дисплея и Пересылки изображения ссылаются на один и тот же буфер кадра.

Необходимо, чтобы частота кадров обновления дисплея была больше, чем частота
35 кадров при пересылке изображения, чтобы избежать отображения частичных изображений. Фиг.90 изображает то, как может происходить фрагментация изображения с медленной частотой обновления изображения дисплея, т.е. обновление изображения дисплея медленнее пересылки изображения.

В изображении, которое содержит комбинацию изображений компьютерной графики и
40 движущихся видеоизображений, видеоданные пикселей могут занимать небольшую часть медиакадра. Это может быть важным в ситуациях, где операция обновления изображения дисплея и пересылка изображения ссылаются на один и тот же буфер кадра. Эти ситуации показаны закрашиванием сетчатым полем на фиг.91, где пиксели, считываемые из буфера для обновления изображения дисплея, могут быть пикселями, записанными в буфер двумя
45 кадрами ранее, или они могут соответствовать кадру, непосредственно записываемому в этот же буфер кадра.

Использование трех буферов кадра в клиенте разрешает проблему малого окна состязания за доступ к буферу кадра, как показано на фиг.92.

Однако существует еще проблема, если частота обновления изображения дисплея
50 меньше частоты медиакадров по линии передачи данных ЦИМД, как показано на фиг.93.

Использование единственного буфера для движущихся изображений видео является в некоторой степени проблематичным, как показано на фиг.94. Если обновление изображения дисплея более быстрое, чем пересылка изображения в буфер, изображение,

обновляемое время от времени, будет показывать верхнюю часть записываемого кадра и нижняя часть изображения будет представлять собой ранее переданный кадр. Если обновление изображения дисплея более быстрое, чем пересылка изображения (предпочтительный режим работы), будут более частые случаи кадров, изображающих подобное разделенное изображение.

XVII. Таблица значений задержки

Пакет параметров задержки на обработку пакета использует функцию табличного поиска для вычисления предсказанной задержки на обработку некоторых команд в клиенте. Значения в таблице увеличиваются логарифмическим образом для получения очень широкого динамического диапазона значений задержки. Примерная таблица значений задержки, полезных для реализации вариантов осуществления изобретения, находится ниже в таблице XX с соответствующими индексными значениями против значений задержки.

Таблица XX

0 - no_delay	37 - 1.5 нс	74 - 51 нс	111 - 1.8 мкс	148 - 62 мкс	185 - 2.2 мс	222 - 75 мс
1 - 46 пс	38 - 1.6 нс	75 - 56 нс	112 - 2.0 мкс	149 - 68 мкс	186 - 2.4 мс	223 - 83 мс
2 - 51 пс	39 - 1.8 нс	76 - 62 нс	113 - 2.2 мкс	150 - 75 мкс	187 - 2.6 мс	224 - 91 мс
3 - 56 пс	40 - 2.0 нс	77 - 68 нс	114 - 2.4 мкс	151 - 83 мкс	188 - 2.9 мс	225 - 100 мс
4 - 62 пс	41 - 2.2 нс	78 - 75 нс	115 - 2.6 мкс	152 - 91 мкс	189 - 3.2 мс	226 - 110 мс
5 - 68 пс	42 - 2.4 нс	79 - 83 нс	116 - 2.9 мкс	153 - 100 мкс	190 - 3.5 мс	227 - 120 мс
6 - 75 пс	43 - 2.6 нс	80 - 91 нс	117 - 3.2 мкс	154 - 110 мкс	191 - 3.8 мс	228 - 130 мс
7 - 83 пс	44 - 2.9 нс	81 - 100 нс	118 - 3.5 мкс	155 - 120 мкс	192 - 4.2 мс	229 - 150 мс
8 - 91 пс	45 - 3.2 нс	82 - 110 нс	119 - 3.8 мкс	156 - 130 мкс	193 - 4.6 мс	230 - 160 мс
9 - 100 пс	46 - 3.5 нс	83 - 120 нс	120 - 4.2 мкс	157 - 150 мкс	194 - 5.1 мс	231 - 180 мс
10 - 110 пс	47 - 3.8 нс	84 - 130 нс	121 - 4.6 мкс	158 - 160 мкс	195 - 5.6 мс	232 - 200 мс
11 - 120 пс	48 - 4.2 нс	85 - 150 нс	122 - 5.1 мкс	159 - 180 мкс	196 - 6.2 мс	233 - 220 мс
12 - 130 пс	49 - 4.6 нс	86 - 160 нс	123 - 5.6 мкс	160 - 200 мкс	197 - 6.8 мс	234 - 240 мс
13 - 150 пс	50 - 5.1 нс	87 - 180 нс	124 - 6.2 мкс	161 - 220 мкс	198 - 7.5 мс	235 - 260 мс
14 - 160 пс	51 - 5.6 нс	88 - 200 нс	125 - 6.8 мкс	162 - 240 мкс	199 - 8.3 мс	236 - 290 мс
15 - 180 пс	52 - 6.2 нс	89 - 220 нс	126 - 7.5 мкс	163 - 260 мкс	200 - 9.1 мс	237 - 320 мс
16 - 200 пс	53 - 6.8 нс	90 - 240 нс	127 - 8.3 мкс	164 - 290 мкс	201 - 10 мс	238 - 350 мс
17 - 220 пс	54 - 7.5 нс	91 - 260 нс	128 - 9.1 мкс	165 - 320 мкс	202 - 11 мс	239 - 380 мс
18 - 240 пс	55 - 8.3 нс	92 - 290 нс	129 - 10 мкс	166 - 350 мкс	203 - 12 мс	240 - 420 мс
19 - 260 пс	56 - 9.1 нс	93 - 320 нс	130 - 11 мкс	167 - 380 мкс	204 - 13 мс	241 - 460 мс
20 - 290 пс	57 - 10 нс	94 - 350 нс	131 - 12 мкс	168 - 420 мкс	205 - 15 мс	242 - 510 мс

21 - 320 пс	58 - 11 нс	95 - 380 нс	132 - 13 мкс	169 - 460 мкс	206 - 16 мс	243 - 560 мс
22 - 350 пс	59 - 12 нс	96 - 420 нс	133 - 15 мкс	170 - 510 мкс	207 - 18 мс	244 - 620 мс
23 - 380 пс	60 - 13 нс	97 - 460 нс	134 - 16 мкс	171 - 560 мкс	208 - 20 мс	245 - 680 мс
24 - 420 пс	61 - 15 нс	98 - 510 нс	135 - 18 мкс	172 - 620 мкс	209 - 22 мс	246 - 750 мс
25 - 460 пс	62 - 16 нс	99 - 560 нс	136 - 20 мкс	173 - 680 мкс	210 - 24 мс	247 - 830 мс
26 - 510 пс	63 - 18 нс	100 - 620 нс	137 - 22 мкс	174 - 750 мкс	211 - 26 мс	248 - 910 мс
27 - 560 пс	64 - 20 нс	101 - 680 нс	138 - 24 мкс	175 - 830 мкс	212 - 29 мс	249 - 1.0 с
28 - 620 пс	65 - 22 нс	102 - 750 нс	139 - 26 мкс	176 - 910 мкс	213 - 32 мс	250 - 1.1 с
29 - 680 пс	66 - 24 нс	103 - 830 нс	140 - 29 мкс	177 - 1.0 мс	214 - 35 мс	251 - 1.2 с
30 - 750 пс	67 - 26 нс	104 - 910 нс	141 - 32 мкс	178 - 1.1 мс	215 - 38 мс	252 - 1.3 с
31 - 830 пс	68 - 29 нс	105 - 1.0 мкс	142 - 35 мкс	179 - 1.2 мс	216 - 42 мс	253 - 1.5 с
32 - 910 пс	69 - 32 нс	106 - 1.1 мкс	143 - 38 мкс	180 - 1.3 мс	217 - 46 мс	254 - 1.6 с
33 - 1.0 нс	70 - 35 нс	107 - 1.2 мкс	144 - 42 мкс	181 - 1.5 мс	218 - 51 мс	255 - бесконечная
34 - 1.1 нс	71 - 38 нс	108 - 1.3 мкс	145 - 46 мкс	182 - 1.6 мс	219 - 56 мс	
35 - 1.2 нс	72 - 42 нс	109 - 1.5 мкс	146 - 51 мкс	183 - 1.8 мс	220 - 62 мс	
36 - 1.3 нс	73 - 46 нс	110 - 1.6 мкс	147 - 56 мкс	184 - 2.0 мс	221 - 68 мс	

Задержка вычисляется посредством выполнения табличного поиска, используя заданный параметр в качестве индекса в таблице. Это означает то, что задержка равна PacketProcessingTable(индекс). Например: если один из параметров из Элемента списка

параметров задержки представляет собой 8-битовое значение, равное 134, тогда задержка равна PacketProcessingTable(134), что составляет 16 мкс. Значение 255 означает, что время завершения команды не может быть определено вычислением и что хост будет проверять Флаги занятости графики в Пакете запроса клиента и состояния или Параметр B7h управления ВПУ НКУМ.

В некоторых случаях эта задержка умножается на высоту, ширину или количество пикселей в изображении назначения и добавляется к другим задержкам для вычисления суммарной задержки на обработку пакета.

XVIII. Поддержка многочисленных клиентов

Текущая версия протокола, по-видимому, не поддерживает непосредственно многочисленные устройства клиента. Однако большинство пакетов содержат зарезервированное поле ИД клиента, которое может использоваться для адресации заданных устройств клиента в системе с многочисленными клиентами. В настоящее время для многих применений это ИД клиента или эти ИД клиента установлены равными нулю.

Пакет заголовка подкадра также содержит поле для указания, поддерживает ли или нет хост систему с многочисленными клиентами. Поэтому существует способ, в котором многочисленные устройства клиента, вероятно, будут подключаться и адресоваться в будущих применениях ЦИМД или протокола, чтобы помочь разработчикам системы планировать будущую совместимость с хостами многочисленных клиентов и клиентами.

В системах, имеющих многочисленных клиентов, полезно, чтобы клиенты подключались к хосту, используя последовательное подключение клиентов, или используя концентраторы, как показано на фиг.95, или используя комбинацию этих методов, как показано на фиг.96. Также может быть полезным для хоста отображать некоторые сообщения об ошибках для управления подключенными клиентами, такие как сообщение об ошибке, когда подключаются один или несколько клиентов, требующих адрес 0, что не должно быть для систем с многочисленными клиентами, так как такие дисплеи, как ожидается, представляют собой единственного клиента или устанавливаются для работы в качестве единственного клиента.

XIX. Приложение

В дополнение к форматам, структурам и содержимому, описанным выше для различных пакетов, используемым для реализации архитектуры и протокола для вариантов осуществления изобретения, здесь представлено более подробное содержание или операции полей для некоторых типов пакетов. Они представлены в данном случае для дополнительного разъяснения их соответствующего использования или операций, чтобы специалисту в данной области техники было легче понять и использовать изобретение для многочисленных применений. Здесь дополнительно описываются только несколько еще неописанных полей. Кроме того, эти поля представлены с примерными определениями и значениями в отношении представленных выше вариантов осуществления. Однако такие значения не должны рассматриваться как ограничения изобретения, но представляют одно или несколько вариантов осуществления, полезных при реализации интерфейса и протокола, и не все варианты осуществления должны вместе или одновременно осуществляться на практике. Другие значения могут использоваться в других вариантах осуществления для достижения требуемого представления данных или результатов пересылки со скоростью передачи данных, что понятно для специалиста в данной области техники.

A. Для пакетов видеопотоков

В одном варианте осуществления поле (2 байта) Атрибутов данных пикселей имеет группу значений битов, которые интерпретируются следующим образом. Биты 1 и 0 выбирают, как направляются данные пикселей дисплея. Для значений битов «11» данные пикселей отображаются обоим глазам или для обоих глаз, для значений битов «10» данные пикселей направляются только левому глазу, и для значений битов «01» данные пикселей направляются только правому глазу, и для значений битов «00» данные пикселей направляются альтернативному дисплею, который может задаваться битами 8-11,

описанными ниже. Если главный дисплей, который находится в клиенте или используется или эксплуатируется клиентом, не поддерживает стереоизображения или формирование изображения в некотором виде, тогда эти команды не могут быть эффективно реализованы, чтобы иметь влияние, требуемое дисплеем. В данной ситуации или

5 конфигурации клиент должен направить данные пикселей на главный дисплей независимо от значений битов или для любой комбинации битов «01», «10» или «11», так как результирующие команды или управление не будет реализовано дисплеем. Рекомендуется, но не требуется вариантами осуществления, чтобы значение «11» использовалось для адресации главного дисплея в тех клиентах, которые не поддерживают
10 возможность отображения стереоизображений.

Бит 2 указывает, присутствуют ли или нет Данные пикселей в чересстрочном формате, при этом значение «0» означает то, что данные пикселей находятся в стандартном последовательном формате и что номер строки (координата Y пикселя) получает приращение 1 при переходе с одной строки на следующую. Когда этот бит имеет значение
15 «1», то данные пикселей находятся в чересстрочном формате и номер строки получает приращение 2 при переходе с одной строки на следующую. Бит 3 указывает, что Данные пикселей находятся в формате чередующихся пикселей. Он аналогичен стандартному чересстрочному режиму, разрешаемому битом 2, но чередование вертикальное вместо горизонтального. Когда Бит 3 равен «0», то Данные пикселей находятся в стандартном
20 последовательном формате и номер столбца (координата X пикселя) получает приращение 1 при приеме каждого последующего пикселя. Когда Бит 3 равен «1», то Данные пикселей находятся в формате чередующихся пикселей и номер столбца получает приращение 2 при приеме каждого пикселя.

Бит 4 указывает, относятся ли или нет данные Пикселей к дисплею или камере, где
25 данные пересылаются на или от внутреннего дисплея для беспроводного телефона или подобного устройства или даже портативного компьютера или таких других устройств, как описано выше, или данные пересылаются на или от камеры, встроенной в или непосредственно связанной с устройством. Когда Бит 4 равен «0», данные Пикселей пересылаются на или от буфера кадра дисплея. Когда Бит 4 равен «1», данные Пикселей
30 пересылаются на или от камеры или устройства видео некоторого типа, причем такие устройства общеизвестны в технике.

Бит 5 используется для указания, когда данные пикселей содержат следующую последующую строку пикселей в дисплее. Это считается случаем, когда Бит 5 устанавливается равным «1». Когда бит 5 устанавливается в «1», тогда параметры Левого
35 края по оси X, Верхнего края по оси Y, Правого края по оси X, Нижнего края по оси Y, Начала по оси X и Начала по оси Y не определяются и игнорируются клиентом. Когда Бит 15 устанавливается на уровень логической единицы, то это указывает, что данные пикселей в этом пакете представляют собой последнюю строку пикселей в изображении. Бит 8 поля Индикаторов возможностей свойств клиента Пакета возможностей клиента
40 указывает, поддерживается ли это свойство.

Биты 7 и 6 представляют собой Биты обновления изображения дисплея, которые задают буфер кадра, где должны записываться данные пикселей. Более конкретные эффекты описаны в другом месте. Для значений битов «01» данные Пикселей записываются в автономный буфер изображения. Для значений битов «00» данные Пикселей записываются
45 в буфер изображения, используемый для обновления изображения дисплея. Для значений битов «11» данные Пикселей записываются во все буферы изображения. Значения битов или комбинация «10» рассматривается как недействительное значение или обозначение, и данные Пикселей игнорируются и не записываются ни в какой из буферов изображения. Это значение может иметь использование для будущих применений интерфейса.

50 Биты 8-11 образуют 4-битовое целое число без знака, которое задает альтернативный дисплей или расположение дисплея, куда должны направляться данные пикселей. Биты 0 и 1 устанавливаются равными «00», чтобы клиент дисплея интерпретировал биты 8-11 в качестве номера альтернативного дисплея. Если биты 0 и 1 не равны «00», тогда биты 8-

11 устанавливаются на уровне логического нуля.

Биты 12-14 резервируются для будущего использования и, как правило, устанавливаются на уровне логического нуля. Бит 15, как описано, используется совместно с битом 5, и установка бита 15 на логическую единицу указывает, что строка
 5 пикселей в поле Данных пикселей является последней строкой пикселей в кадре данных. Следующий Пакет видеопотока, бит 5 которого установлен на логическую единицу, будет соответствовать первой строке пикселей следующего видеокадра.

2-байтовые поля Начала по оси X и Начала по оси Y задают абсолютные координаты X и Y точки (Начало по оси X, Начало по оси Y) для первого пикселя в поле Данных
 10 пикселей. 2-байтовые поля Левого края по оси X и Верхнего края по оси Y задают координату X левого края и координату Y верхнего края окна экрана, заполненного полем Данных пикселей, тогда как Правый край по оси X и Нижний край по оси Y задают координату X правого края и координату Y нижнего края обновляемого окна.

Поле (2 байта) Числа пикселей задает количество пикселей в поле Данных пикселей
 15 ниже.

Поле (2 байта) ЦИК параметров содержит ЦИК всех байтов от Длины пакета до Числа пикселей. Если проверка этим ЦИК unsuccessful, тогда весь пакет отбрасывается.

Поле Данных пикселей содержит необработанную видеoinформацию, которая должна быть отображена и которая форматируется так, как описывается полем Дескриптора
 20 формата видеоданных. Данные передаются одной «строкой» за раз, как описано в другом месте. Когда Бит 5 поля Атрибутов данных пикселей установлен в единицу логического уровня, тогда поле Данных пикселей содержит точно одну строку пикселей, причем первый пиксель передается соответствующим самому левому пикселю и последний пиксель передается соответствующим самому правому пикселю.

Поле (2 байта) ЦИК Данных пикселей содержит 16-битовый ЦИК только Данных пикселей. Если проверка посредством ЦИК этого значения unsuccessful, тогда Данные пикселей могут все же использоваться, но число ошибок ЦИК получает приращение.

В. Для пакетов аудиопотока

В одном варианте осуществления поле (1 байт) ИД аудиоканала использует значение 8-
 30 битового целого числа без знака для идентификации конкретного аудиоканала, на который посылаются аудиоданные устройством клиента. Физические аудиоканалы задаются или отображаются на физические каналы посредством этого поля в качестве значений 0, 1, 2, 3, 4, 5, 6 или 7, которые указывают левый передний, правый передний, левый задний, правый задний, передний центральный, сверхнизкочастотный, объемный левый и
 35 объемный правый каналы соответственно. Значение 254 в поле ИД аудиоканала указывает на то, что один поток цифровых аудиоотсчетов посылается как на левый передний, так и на правый передний каналы. Это упрощает обмен данными для применений, таких как, где стереофонический головной телефон используется для передачи речи, приложений повышения эффективности используются на ПЦП или других применений, где простой
 40 Пользовательский интерфейс создает предупреждающие тональные сигналы. Значения для поля ИД, находящиеся в диапазоне 8-253 и равные 255, в настоящее время резервируются для использования там, где новые разработки потребуют дополнительных обозначений, что предусматривается специалистами в данной области техники.

Поле (1 байт) Резервированное 1, как правило, резервируется для будущего
 45 использования и имеет все биты этого поля, установленные в ноль. Одним назначением этого поля является то, чтобы вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

Поле (2 байта) Числа аудиоотсчетов задает количество аудиоотсчетов в этом пакете.

Поле Битов на отсчет и упаковки содержит 1 байт, который задает формат упаковки
 50 аудиоданных. В одном варианте осуществления обычно используемым форматом является такой, что Биты 4-0 определяют количество битов на ИКМ-аудиоотсчет. Бит 5 тогда задает, упакованы ли или нет отсчеты Цифровых аудиоданных. Как упомянуто выше,

фиг.12 иллюстрирует различие между упакованными и выровненными по байтам аудиоотсчетами. Значение «0» Бита 5 указывает, что каждый ИКМ-аудиоотсчет в поле Цифровых аудиоданных выровнен по байтам с границей байтов интерфейса, и значение «1» указывает, что каждый последующий ИКМ-аудиоотсчет упакован к предыдущему аудиоотсчету. Этот бит действует только тогда, когда значение, определенное в битах 4-0 (количество битов на ИКМ-аудиоотсчет), не является кратным восьми. Биты 7-6 резервируются для использования там, где системные разработки требуют дополнительных обозначений и, как правило, устанавливаются на значение нуля.

Поле (1 байт) Частоты аудиоотсчетов задает частоту ИКМ-аудиоотсчетов.

Используемым форматом является такой, что значение 0 указывает частоту 8000 отсчетов в секунду (отсчет/с), значение 1 указывает 16000 отсчет/с, значение 2 - 24000 отсчет/с, значение 3 - 32000 отсчет/с, значение 4 - 40000 отсчет/с, значение 5 - 48000 отсчет/с, значение 6 - 11025 отсчет/с, значение 7 - 22050 отсчет/с и значение 8 - 44100 отсчет/с соответственно, причем значения 9-255 резервируются для будущего применения, поэтому в настоящее время они устанавливаются в нуль.

Поле (2 байта) ЦИК параметров содержит 16-битовый ЦИК всех байтов от Длины пакета до Частоты аудиоотсчетов. Если проверка этим ЦИК была неуспешной, тогда весь пакет отбрасывается. Поле Цифровых аудиоданных содержит необработанные аудиоотсчеты, подлежащие воспроизведению, и представлены обычно в виде линейного формата в качестве целых чисел без знака. Поле (2 байта) ЦИК аудиоданных содержит 16-битовый ЦИК только для Аудиоданных. Если проверка этим ЦИК неуспешна, тогда Аудиоданные все же могут использоваться, но число ошибок ЦИК получает приращение.

С. Для пакетов определяемого пользователем потока

В одном варианте осуществления 2-байтовое поле Номера ИД потока используется для идентификации конкретного определяемого пользователем потока. Содержимое полей Параметров потока и Данных потока обычно определяются изготовителем оборудования ЦИМД. 2-байтовое поле ЦИК параметров потока содержит 16-битовый ЦИК всех байтов параметров потока, начиная с Длины пакета и до байта Кодирование аудио. Если проверка этим ЦИК неуспешна, тогда весь пакет отбрасывается. Оба поля Параметров потока и ЦИК параметров потока могут отбрасываться, если они не нужны для конечного применения ЦИМД, т.е. они считаются необязательными. 2-байтовое поле ЦИК данных потока содержит ЦИК только Данных потока. Если проверка этим ЦИК неуспешна, тогда использование Данных потока является необязательным в зависимости от требований применения. Использование данных потока, зависящее от ЦИК без ошибок, как правило, требует, чтобы данные потока были буферизованы до тех пор, пока не будет подтверждено, что ЦИК без ошибок. Число ошибок ЦИК получает приращение, если проверка ЦИК не проходит.

Д. Для пакетов карты цветов

2-байтовое поле ИД hClient содержит информацию или значения, которые резервируются для ИД клиента, используемые, как указано ранее. Так как это поле, как правило, резервируется для будущего использования, текущее значение устанавливается на нуль посредством установки битов в «0».

2-байтовое поле Числа элементов карты цветов использует значения для задания общего количества 3-байтовых элементов карты цветов, которые содержатся в поле Данных карты цветов, или элементов таблицы карты цветов, которые существуют в Данных карты цветов в этом пакете. В данном варианте осуществления количество байтов в Данных карты цветов в 3 раза больше Числа элементов карты цветов. Число элементов карты цветов устанавливается равным нулю при отсутствии посылки данных карты цветов. Если Размер карты цветов равен нулю, тогда значение Смещения карты цветов, как правило, все же посылается, но игнорируется дисплеем. Поле (4 байта) Смещения карты цветов задает смещение Данных карты цветов в этом пакете с начала таблицы карты цветов в устройстве клиента.

2-байтовое поле ЦИК параметров содержит ЦИК всех байтов с Длины пакета до байта Кодирования аудио. Если проверка этим ЦИК неуспешна, тогда отбрасывается весь пакет.

Для поля Данных карты цветов, разрядность каждой ячейки карты цветов представляет собой задаваемое полем Размера элемента карты цветов, где в одном варианте осуществления первая часть задает величину синего цвета, вторая часть задает величину зеленого цвета, и третья часть задает величину красного цвета. Поле Размера карты

5 цветов задает количество 3-байтовых элементов таблицы карты цветов, которые существуют в поле Данных карты цветов. Если одна карта цветов не может разместиться в одном пакете Формата видеоданных и карты цветов, тогда вся карта цветов может задаваться посылкой многочисленных пакетов с различными Данными карты цветов и Смещениями карты цветов в каждом пакете. Количество битов синего, зеленого и красного

10 цветов в каждом элементе данных карты цветов, как правило, одинакова с заданным в поле Разрядности карты цветов формата RGB Пакета возможностей дисплея.

2-байтовое поле ЦИК данных карты цветов содержит ЦИК только Данных карты цветов. Если проверка этим ЦИК неуспешна, тогда Данные карты цветов все же могут использоваться, но число ошибок ЦИК получает приращение.

15 Каждый элемент данных карты цветов должен передаваться в следующем порядке: синий, зеленый, красный, причем самый младший бит каждой составляющей передается первым. Индивидуальные составляющие красного, зеленого и синего цвета каждого элемента карты цветов упаковываются, но каждый элемент карты цветов (самый младший бит составляющей синего цвета) должен выравниваться по байтам. Фиг.97 иллюстрирует

20 пример элементов данных карты цветов с 6 битами синего цвета, 8 битами зеленого цвета и 7 битами красного цвета. Для этого примера Размер элемента карты цветов в Пакете карты цветов равен 21 и поле Разрядности карты цветов формата RGB Пакета возможностей клиента равно 0x0786.

Е. Для пакетов инкапсуляции обратной линии передачи данных

25 Поле (2 байта) ЦИК параметров содержит 16-битовый ЦИК всех байтов с Длины пакета до Длины реверсирования передачи. Если проверка этим ЦИК неуспешна, тогда отбрасывается весь пакет.

В одном варианте осуществления поле (1 байт) Флагов обратной линии передачи данных содержит группу флагов для запроса информации от клиента и задает тип

30 обратной линии передачи данных. Если бит (например, Бит 0) установлен на уровень логической единицы, тогда хост запрашивает заданную информацию у клиента, но если бит установлен на уровень логического нуля, тогда хосту не требуется информация от клиента. Бит 0 используется для указания, когда хосту требуется Пакет возможностей клиента, который, как правило, посылается клиентом хосту в поле Пакетов обратных

35 данных. Бит 1 используется для указания, когда хосту требуется Пакет запроса клиента и состояния, который посылается клиентом хосту в поле Пакетов обратных данных. Остальные биты (в данном случае биты 2-7) резервируются для будущего использования и устанавливаются в нуль. Однако может использоваться больше битов, когда требуется, для установки флагов для обратной линии передачи данных.

40 Поле (1 байт) Делителя обратной скорости передачи задает количество циклов MDDI_Stb, которые происходят относительно тактовых импульсов данных обратной линии передачи данных. Частота тактовых импульсов данных обратной линии передачи данных равна частоте тактовых импульсов данных прямой линии передачи данных, деленной на два, умноженное на Делитель обратной скорости передачи. Скорость передачи данных по

45 обратной линии передачи данных связана с частотой тактовых импульсов данных обратной линии передачи данных и Типом интерфейса на обратной линии передачи данных. В данном варианте осуществления для интерфейса Типа 1 обратная скорость передачи данных равна частоте тактовых импульсов данных обратной линии передачи данных, для интерфейсов Типа 2, Типа 3 и Типа 4 обратные скорости передачи данных равны частоте

50 тактовых импульсов данных обратной линии передачи данных, умноженной соответственно в два, четыре и восемь раз.

Поле Всех нулей 1 содержит группу байтов, в данном случае 8, которые устанавливаются равными нулю по значению посредством установки битов на уровень

логического нуля, и используется для обеспечения того, чтобы все сигналы MDDI_Data находились на уровне логического нуля в течение достаточного времени, чтобы дать возможность клиенту начать восстановление тактовых импульсов, используя только MDDI_Stb, перед выключением линейных драйверов хоста в течение поля Реверсирования

5 передачи 1. В одном варианте осуществления длина поля Всех нулей 1 больше или равна количеству байтовых передач по прямой линии передачи данных, умноженному на задержку на двойное прохождение кабеля.

Поле (1 байт) Длины реверсирования передачи 1 задает общее количество байтов, которое выделено для Реверсирования передачи 1 для установления первого периода

10 реверсирования передачи. Поле Реверсирования передачи 1 использует количество байтов, задаваемое параметром Длины реверсирования передачи 1, которые выделены для того, чтобы выполнить включение линейных драйверов MDDI_Data в клиенте, до того как выключатся линейные драйвера в хосте. Клиент включает свои линейные драйверы MDDI_Data в течение бита 0 Реверсирования передачи 1, и хост выключает свои выходы

15 так, чтобы они были полностью выключенными до последнего бита Реверсирования передачи 1. Сигнал MDDI_Stb действует так, как если бы MDDI_Data0 были на уровне логического нуля в течение всего периода Реверсирования передачи 1. Более полное описание установок Реверсирования передачи 1 приведено выше.

Поле Пакетов обратных данных содержит группу пакетов данных, пересылаемых с

20 клиента на хост. Клиент может посылать пакеты заполнителя или возбуждать линии MDDI_Data состоянием или уровнем логического нуля, когда у него нет данных для посылки хосту. В данном варианте осуществления, если линии MDDI_Data возбуждаются нулем, хост будет интерпретировать это как пакет с нулевой длиной (недействительная длина), и хост не будет принимать дополнительные пакеты от клиента в течение текущего

25 Пакета инкапсуляции обратной линии передачи данных.

Поле (1 байт) Длины реверсирования передачи 2 задает общее количество байтов, которые выделяются для Реверсирования передачи 2, для установления второго периода реверсирования передачи. Рекомендуемая длина Реверсирования передачи 2 равна

30 количеству байтов, требуемых для задержки на двойное прохождение плюс время, требуемое для того, чтобы хост включил свои драйверы MDDI_Data. Длина Реверсирования передачи 2 также может представлять собой значение, которое больше минимально требуемого или вычисленного значения, чтобы предоставить достаточное время для обработки пакетов обратной линии передачи данных в хосте.

Поле Реверсирования передачи 2 состоит из количества байтов, задаваемых

35 параметром Длины реверсирования передачи. Хост ожидает в течение по меньшей мере времени задержки на двойное прохождение до включения своих линейных драйверов MDDI_Data в течение Реверсирования передачи 2. Хост включает свои линейные драйверы MDDI_Data, так что они в основном будут полностью включены до последнего бита Реверсирования передачи 2, и клиент выключает свои выходы, так что они в основном

40 будут полностью выключены перед последним битом Реверсирования передачи 2. Назначением поля Реверсирования передачи 2 является предоставление возможности передачи или пересылки с клиента оставшегося количества данных из поля Пакетов обратных данных. Из-за изменений в различных системах, реализующих интерфейс, и величины выделенного запаса надежности, возможно, что ни хост, ни клиент не будут

45 возбуждать сигналы MDDI_Data уровнем логического нуля в течение некоторых частей периода поля Реверсирования передачи 2, наблюдаемым линейными приемниками в или на хосте. Сигнал MDDI_Stb действует так, как если бы MDDI_Data0 был на уровне логического нуля в течение по существу всего периода Реверсирования передачи 2. Описание установок Реверсирования передачи 2 приведено выше.

Поле Пакетов обратных данных содержит группу пакетов данных, пересылаемых с

50 клиента на хост. Как указано ранее, пакеты Заполнителя посылаются для заполнения оставшегося пространства, которое не используется пакетами других типов.

Поле Всех нулей 2 содержит группу байтов (8 в данном варианте осуществления),

которые устанавливаются равными нулю по значению посредством установки битов на уровень логического нуля, и используется для обеспечения того, чтобы все сигналы MDDI_Data находились на уровне логического нуля в течение достаточного времени, чтобы предоставить возможность клиенту начать восстановление тактовых импульсов, используя как MDDI_Data0, так и MDDI_Stb, после включения линейных драйверов хоста после поля Реверсирования передачи 2.

F. Для пакетов возможностей клиента

Как изображено для одного варианта осуществления поле Версии протокола использует 2 байта для задания версии протокола, используемой клиентом. Первоначальная версия в настоящее время устанавливается равной единице и будет изменяться во времени, когда будут создаваться новые версии, что будет известно, в то время как поле Минимальной версии протокола использует 2 байта для задания минимальной версии протокола, которую клиент может использовать или интерпретировать. В данном случае нулевое значение также является действительным значением. Поле (2 байта) Возможностей скорости передачи данных задает максимальную скорость передачи данных, с которой клиент может принимать по каждой паре данных по прямой линии передачи данных интерфейса и которая задается в виде мегабитов в секунду (Мбит/с). Поле (1 байт) Возможностей типа интерфейса задает типы интерфейса, которые поддерживаются на прямой и обратной линиях передачи данных. Бит, установленный в «1», указывает, что поддерживается заданный тип интерфейса, и бит, установленный в «0», указывает, что заданный тип не поддерживается. Хосты и клиенты должны поддерживать по меньшей мере Тип 1 на прямой и обратной линиях передачи данных. Нет требования на поддержку непрерывного диапазона типов интерфейса. Например, будет совершенно действительной поддержка только Типа 1 и Типа 3, но не Типа 3 и Типа 4 в интерфейсе. Также не является необходимым, чтобы прямая и обратная линии передачи данных работали с одинаковым типом интерфейса. Однако когда линия передачи данных выходит из «спячки», как прямая, так и обратная линия передачи данных должны начинать работу в режиме Типа 1 до тех пор, пока другие режимы не смогут быть согласованы, выбраны или иным образом одобрены для использования как хостом, так и клиентом.

Поддерживаемые интерфейсы указываются в одном варианте осуществления посредством выбора Бита 0, Бита 1 или Бита 2 для выбора режима или Типа 2 (2 бит), или Типа 3 (4 бит), или Типа 4 (8 бит) на прямой линии передачи данных соответственно; и Бита 3, Бита 4 или Бита 5 для выбора режима или Типа 2, или Типа 3, или Типа 4 на обратной линии передачи данных соответственно; при этом Биты 6 и 7 резервируются и, как правило, устанавливаются в нуль в настоящее время. Поля Ширины и высоты растрового изображения, в данном случае каждое имеет длину 2 байта, задают ширину и высоту растрового изображения соответственно в пикселях.

Поле (1 байт) Возможностей монохромного формата используется для задания количества битов разрешения, которое может отображаться в монохромном формате. Если дисплей не может использовать монохромный формат, тогда это значение устанавливается в нуль. Биты 7-4 резервируются для будущего использования и, таким образом, устанавливаются в нуль. Биты 3-0 определяют максимальное количество битов серой шкалы, которое может существовать для каждого пикселя. Эти четыре бита делают возможным задавать значения от 1 до 15 для каждого пикселя. Если значение равно нулю, тогда монохромный формат не поддерживается дисплеем.

Поле Возможностей байеровского формата использует 1 байт для задания количества битов разрешения, группы пикселей и порядка пикселей, которые могут пересылаться в байеровском формате. Если клиент не может использовать байеровский формат, тогда это значение равно нулю. Поле Возможностей байеровского формата состоит из следующих значений: Биты 3-0 определяют максимальное количество битов яркости, которое существует в каждом пикселе, тогда как Биты 5-4 определяют комбинацию группы пикселей, которая требуется, тогда как Биты 8-6 определяют порядок пикселей, который требуется; причем Биты 14-9 резервируются для будущего использования и, как правило,

устанавливаются тем временем в нуль. Бит 15, когда он установлен на уровень логической единицы, указывает, что клиент может принимать данные пикселей байеровского формата или в упакованном, или в неупакованном формате. Если бит 15 установлен в нуль, то это указывает, что клиент может принимать данные пикселей

5 байеровского формата только в неупакованном формате.

Поле (3 байта) Возможностей карты цветов задает максимальное количество элементов таблицы, которые имеются в таблице карты цветов в дисплее. Если дисплей не может использовать формат карты цветов, тогда это значение устанавливается в нуль.

Поле (2 байта) Возможностей формата RGB задает количество битов разрешения, которое может отображаться в формате RGB. Если дисплей не может использовать формат RGB, тогда это значение равно нулю. Слово Возможностей формата RGB состоит из трех отдельных значений без знака, где: Биты 3-0 определяют максимальное количество битов синего цвета, Биты 7-4 определяют максимальное количество битов зеленого цвета, и Биты 11-8 определяют максимальное количество битов красного цвета в каждом пикселе. В настоящее время Биты 14-12 резервируются для будущего использования и, как правило, устанавливаются в нуль. Биты 14-12 резервируются для будущего использования и, как правило, устанавливаются в нуль. Бит 15, когда он установлен на уровень логической единицы, указывает, что клиент может принимать RGB-данные пикселей или в упакованном, или в неупакованном формате. Если бит 15 установлен на уровень логического нуля, то это указывает на то, что клиент может принимать RGB-данные пикселей только в неупакованном формате.

Поле (2 байта) Возможностей формата Y Cr Cb задает количество битов разрешения, которое может отображаться в формате Y Cr Cb. Если дисплей не может использовать формат Y Cr Cb, тогда это значение устанавливается равным нулю. Слово Возможностей формата Y Cr Cb состоит из трех отдельных значений без знака, где: Биты 3-0 определяют максимальное количество битов в отсчете Cb, Биты 7-4 определяют максимальное количество битов в отсчете Cr, биты 11-8 определяют максимальное количество битов в отсчете Y, и биты 15-12 в настоящее время резервируются для будущего использования и устанавливаются в нуль.

Поле Возможностей свойств клиента использует 4 байта, которые содержат группу флагов, указывающих заданные свойства в клиенте, которые поддерживаются. Бит, установленный на уровень логической единицы, указывает, что возможность поддерживается, тогда как бит, установленный на уровень логического нуля, указывает, что возможность не поддерживается. В одном варианте осуществления значение для Бита 0 указывает, поддерживается ли или нет Пакет пересылки блоков растрового изображения (пакет типа 71). Значение для битов 1, 2 и 3 указывают, поддерживаются ли или нет соответственно, Пакет заливки растровой области (пакет типа 72), Пакет заливка растровым узором (пакет типа 73) или Пакет канала данных линии обмена данными (пакет типа 74). Значение для Бита 4 указывает, имеет ли или нет клиент возможность выполнения одного цвета прозрачным, используя Пакет разрешения прозрачного цвета, тогда как значения для Битов 5 и 6 указывают, может ли клиент принимать видеоданные или аудиоданные в упакованном формате соответственно, и значение для Бита 7 указывает, может ли или нет клиент посылать видеопоток по обратной линии передачи данных с камеры. Значение для Бита 8 указывает, имеет ли или нет клиент возможность приема полной строки данных пикселей и игнорирования адресации дисплея, как задается битом 5 поля Атрибутов данных пикселей Пакета видеопотока, и клиент также может обнаруживать кадровую синхронизацию или окончание данных видеокadra, используя бит 15 Поля атрибутов данных пикселей.

Значение для Битов 11 и 12 указывают, когда клиент находится в состоянии обмена данными или с указательным устройством, и может посылать и принимать Пакеты данных указательного устройства, или с клавиатурой, и может посылать и принимать Пакеты данных клавиатуры соответственно. Значение для Бита 13 указывает, имеет ли или нет клиент возможность устанавливать один или несколько параметров аудио или видео

посредством поддержки пакетов Свойств ВПУ: Пакет запроса свойств ВПУ, Пакет ответа свойств ВПУ, Пакет установки свойств ВПУ, Пакет запроса действительных параметров и Пакет ответа действительных параметров. Значение для Бита 14 указывает, имеет ли или нет клиент возможность записи данных пикселей в автономный буфер кадра дисплея. Если
 5 этот бит устанавливается на уровень логической единицы, тогда Биты обновления изображения дисплея (биты 7 и 6 поля Атрибутов данных пикселей Пакета видеопотока) могут устанавливаться на значения «01».

Значение для Бита 15 указывает, когда клиент имеет возможность записи данных пикселей только в буфер кадра дисплея, используемый в настоящий момент времени для
 10 обновления изображения дисплея. Если этот бит установлен в единицу, тогда Биты обновления изображения дисплея (биты 7 и 6 поля Атрибутов данных пикселей Пакета видеопотока) могут устанавливаться на значения «00». Значение для Бита 16 указывает, когда клиент имеет возможность записи данных пикселей из одного Пакета видеопотока во все буферы кадра дисплея. Если этот бит установлен в единицу, тогда Биты обновления
 15 изображения дисплея (биты 7 и 6 поля Атрибутов данных пикселей Пакета видеопотока) могут устанавливаться в значение «11».

Значение для Бита 17 указывает, когда клиент имеет возможность ответа на Пакет запроса заданного состояния, значение для Бита 18 указывает, когда клиент имеет возможность ответа на Пакет измерения задержки на двойное прохождение, и значение
 20 для Бита 19 указывает, когда клиент имеет возможность ответа на Пакет калибровки перекося прямой линии передачи данных.

Значение для Бита 20 указывает, когда клиент имеет возможность ответа на Пакет состояния питания дисплея.

Значение для Бита 21 указывает, когда клиент имеет возможность интерпретировать
 25 Пакет запроса заданного состояния и отвечать Пакетом списка ответа действительных состояний. Клиент указывает возможность возврата дополнительного состояния в поле Списка ответа действительных параметров Пакета списка ответа действительных состояний, как описано в другом месте.

Значение для Бита 22 указывает, имеет ли или нет клиент возможность ответа на Пакет
 30 доступа к регистрам. Биты 9-10 и 23-31 в настоящее время резервируются для будущего использования или альтернативных обозначений, полезных для разработчиков системы, и, как правило, устанавливаются в нуль.

Поле (1 байт) Возможностей частоты видеокадров дисплея задает максимальную
 35 возможность обновления видеокадров дисплея в кадрах в секунду. Хост может выбирать для обновления изображения меньшую частоту, чем значение, заданное в этом поле.

Поле (2 байта) Емкости буфера аудио задает емкость буфера регулируемой емкости в Дисплее, который выделяется для каждого аудиопотока.

Поле (2 байта) Возможностей аудиоканала содержит группу флагов, которые указывают, какие аудиоканалы поддерживаются клиентом или подключенным к клиенту устройством.

40 Бит, установленный в единицу, указывает, что канал поддерживается, и бит, установленный в нуль, указывает, что канал не поддерживается. Позиции Битов присваиваются различным каналам, например, позиции Битов 0, 1, 2, 3, 4, 5, 6 и 7 в одном варианте осуществления указывают левый передний, правый передний, левый задний, правый задний, передний центральный, сверхнизкочастотный, объемный левый и
 45 объемный правый каналы соответственно. Биты 8-14 в настоящее время резервируются для будущего использования и, как правило, устанавливаются в нуль. В одном варианте осуществления Бит 15 используется для указания, обеспечивает ли клиент поддержку Пакета включения прямых аудиоканалов. Если это так, Бит 15 устанавливается на уровень логической единицы. Если, однако, клиент не может выключать аудиоканалы под
 50 действием Пакета включения прямых аудиоканалов или если клиент не поддерживает никакую возможность аудио, тогда этот бит устанавливается на уровень или значение логического нуля.

2-байтовое поле Возможностей частоты аудиоотсчетов для прямой линии передачи

данных содержит группу флагов для указания возможностей частоты аудиоотсчетов устройства клиента. Позиции битов присваиваются различным частотам соответственно, так что Битам 0, 1, 2, 3, 4, 5, 6, 7 и 8 присваиваются 8000, 16000, 24000, 32000, 40000, 48000, 11025, 22050 и 44100 отсчетов в секунду (отсчет/с) соответственно,

5 причем биты 9-15 резервируются для будущего или альтернативного использования частоты, когда потребуется, так что в настоящее время они устанавливаются в «0». Установка значения бита для одного из этих битов в «1» указывает, что поддерживается эта конкретная частота отсчетов, и установка бита в «0» указывает, что эта частота отсчетов не поддерживается.

10 Поле (2 байта) Минимальной частоты подкадров задает минимальную частоту подкадров в кадрах в секунду. Минимальная частота подкадров обеспечивает частоту обновления состояния дисплея, достаточную для считывания определенных датчиков или указательных устройств в клиенте.

2-байтовое поле Возможностей частоты отсчетов микрофона для обратной линии
15 передачи данных содержит группу флагов, которые указывают возможности частоты аудиоотсчетов микрофона в устройстве клиента. Для целей ЦИМД микрофон устройства клиента конфигурируется на поддержку как минимум частоты по меньшей мере 8000 отсчетов в секунду. Позиции битов для этого поля присвоены различным частоты, при этом позиции битов 0, 1, 2, 3, 4, 5, 6, 7 и 8, например, используются для того, чтобы
20 представлять 8000, 16000, 24000, 32000, 40000, 48000, 11025, 22050 и 44100 отсчетов в секунду (отсчет/с) соответственно, причем Биты 9-15 резервируются для будущих или альтернативных применений частоты, когда потребуется, поэтому они в настоящее время устанавливаются в «0». Установка значения бита для одного из этих битов в «1» указывает, что поддерживается эта конкретная частота отсчетов, и установка бита в «0»
25 указывает, что эта частота отсчетов не поддерживается. Если микрофон не подсоединен, тогда каждый бит Возможностей частоты отсчетов микрофона устанавливается равным нулю.

Поле (в данном случае 1 байт) Формата данных клавиатуры задает, подключена ли или нет клавиатура к системе клиента, и тип клавиатуры, которая подключена. В одном
30 варианте осуществления значение, установленное Битами 6-0, используется для определения типа клавиатуры, которая подключена. Если значение равно нулю (0), тогда тип клавиатуры считается неизвестным. Для значения 1 считается, что формат данных клавиатуры представляет собой стандартный тип PS/2. В настоящее время значения в диапазоне 2-125 не используются и резервируются для использования разработчиками системы и специалистами по встраиванию интерфейса или разработчиками продукта для
35 определения заданной клавиатуры или устройств ввода для использования с ЦИМД и соответствующими клиентами или хостами. Значение 126 используется для указания, что формат данных клавиатуры является определяемым пользователем, тогда как значение 127 используется для указания, что клавиатура не может быть подключена к этому
40 клиенту. Кроме того, Бит 7 может использоваться для указания, может ли или нет клавиатура обмениваться данными с клиентом. Предназначенным использованием этого бита является указание, когда клавиатура может обмениваться данными с клиентом, используя беспроводную линию передачи данных. Бит 7 устанавливается на нулевой уровень, если биты 6-0 указывают, что клавиатура не может быть подключена к клиенту.
45 Поэтому для одного варианта осуществления, когда значение Бита 7 равно 0, клавиатура и клиент не могут обмениваться данными, тогда как, если значение Бита 7 равно 1, клавиатура и клиент подтвердили, что они могут обмениваться данными друг с другом.

Поле (в данном случае 1 байт) Формата данных указательного устройства задает, подключено ли или нет указательное устройство к системе клиента, и тип указательного
50 устройства, которое подключено. В одном варианте осуществления значение, установленное Битами 6-0, используется для определения типа указательного устройства, которое подключено. Если значение равно нулю (0), тогда тип указательного устройства считается неизвестным. Для значения 1 считается, что формат данных указательного

устройства представляет собой стандартный тип PS/2. В настоящее время значения в диапазоне 2-125 не используются и резервируются для использования разработчиками системы и специалистами по встраиванию интерфейса или разработчиками продукта для определения заданного указательного устройства или устройств ввода для использования с ЦИМД и соответствующими клиентами или хостами. Значение 126 используется для указания, что формат данных указательного устройства является определяемым пользователем, тогда как значение 127 используется для указания, что указательное устройство не может быть подключено к этому клиенту. Кроме того, Бит 7 может использоваться для указания, может ли или нет указательное устройство обмениваться данными с клиентом. Предназначенным использованием этого бита является указание, когда клавиатура может обмениваться данными с клиентом, используя беспроводную линию передачи данных. Бит 7 устанавливается на нулевой уровень, если биты 6-0 указывают, что указательное устройство не может быть подключено к клиенту. Поэтому для одного варианта осуществления, когда значение Бита 7 равно 0, указательное устройство и клиент не могут обмениваться данными, тогда как, если значение Бита 7 равно 1, указательное устройство и клиент подтвердили, что они могут обмениваться данными друг с другом.

Поле (2 байта) Типа защиты содержимого содержит группу флагов, которые указывают тип защиты цифрового содержимого, которая поддерживается Дисплеем. В настоящее время позиция 0 битов используется для указания того, когда поддерживается ЗЦСП, и позиция 1 битов используется для указания того, когда поддерживается ШСЗЦС, причем позиции 2-15 битов зарезервированы для использования с другими схемами защиты, когда потребуется или будут в наличии, так что они в настоящее время устанавливаются в нуль.

Поле (в данном случае 2 байта) Названия изготовителя содержит 3-знаковый ИД РАПС изготовителя, упакованный в три 5-битовых знака аналогично тому как в спецификации РДИД АСВЭ. Знак «А» представлен как двоичное 00001, знак «Z» представлен как двоичное 11010, и все буквы между «А» и «Z» представлены как последовательные двоичные значения, которые соответствуют алфавитной последовательности между «А» и «Z». Самый старший бит поля Названия изготовителя не используется и, как правило, устанавливается в логический нуль в настоящее время до тех пор, пока не будет использоваться для будущих реализаций. Например, изготовитель, представленный строкой «XYZ», будет иметь значение Названия изготовителя 0x633a. Если это поле не поддерживается клиентом, оно, как правило, устанавливается в нуль. Поле Кода продукта использует 2 байта и содержит код продукта, присвоенный изготовителем дисплея. Если это поле не поддерживается клиентом, оно, как правило, устанавливается в нуль.

Поля (в данном случае 2 байта каждый) Резервированное 1, Резервированное 2 и Резервированное 3 резервируются для будущего использования для сообщения информации. Все биты в этих полях, как правило, устанавливаются на уровень логического нуля. Назначение таких полей в настоящее время заключается в том, чтобы вызывать выравнивание всех последующих 2-байтовых полей с адресом 16-битовых слов и вызывать выравнивание 4-байтовых полей с адресом 32-битовых слов.

Поле Серийного номера использует 4 байта в данном варианте осуществления для задания серийного номера дисплея в числовом виде. Если это поле не поддерживается клиентом, оно, как правило, устанавливается в нуль. Поле Недели изготовления использует 1 байт для определения недели изготовления дисплея. Это значение обычно находится в диапазоне 1-53, если оно поддерживается клиентом. Если это поле не поддерживается клиентом, оно устанавливается в нуль. Поле Года изготовления представляет собой 1 байт, который определяет год изготовления дисплея. Это значение представляет собой смещение с года 1990. Годы в диапазоне 1991-2245 могут выражаться этим полем. Например: год 2003 соответствует значению Года изготовления 13. Если это поле не поддерживается клиентом, оно устанавливается в нуль.

Поле (в данном случае 2 байта) ЦИК содержит 16-битовый ЦИК всех байтов в пакете, включая Длину пакета.

Г. Для пакетов запроса клиента и состояния

Поле (3 байта) Запроса обратной линии передачи данных задает количество байтов, которые необходимы клиенту на обратной линии передачи данных в следующем подкадре для посылки информации на хост.

5 Поле (1 байт) Числа ошибок ЦИК указывает, сколько произошло ошибок ЦИК с начала медиакадра. Число ошибок ЦИК сбрасывается тогда, когда посылается пакет заголовка подкадра с Числом подкадров, равным нулю. Если фактическое количество ошибок ЦИК превышает 255, то тогда это значение, как правило, ограничивается значением 255.

Поле Изменения возможностей использует 1 байт для указания изменения в
10 возможностях клиента. Это может произойти, если пользователь подключает периферийное устройство, такое как микрофон, клавиатуру или дисплей, или по некоторой другой причине. Когда Биты [7:0] равны 0, тогда возможности не изменились после посылки последнего Пакета возможностей клиента. Однако когда Биты [7:0] равны 1-255, возможности изменились. Пакет возможностей клиента исследуется с целью определения
15 новых характеристик дисплея.

Поле Флагов занятости клиента использует 2 байта для указания, что клиент выполняет заданную функцию и не готов еще принимать другой пакет, относящийся к этой функции. Бит, установленный на уровень или значение логической единицы, указывает, что конкретная функция в настоящий момент выполняется клиентом и что относящаяся
20 функция в клиенте находится в состоянии занятости. Если относящаяся функция в клиенте находится в состоянии готовности, бит устанавливается в логический ноль. Клиент должен возратить состояние занятости (бит, установленный в единицу) для всех функций, которые не поддерживаются в клиенте.

В одном варианте осуществления эти биты интерпретируются в соответствии с
25 зависимостью: если Бит 0 равен «1», тогда функция пересылки блоков растрового изображения находится в состоянии занятости, тогда как, если Бит 1 равен «1», тогда функция заливки растровой области находится в состоянии занятости, и, если Бит 2 равен «1», тогда функция заливки растровым узором находится в состоянии занятости. В настоящее время Биты 3-15 остаются зарезервированными для будущего использования и,
30 как правило, устанавливаются на уровень или в состояние логической единицы для указания состояния занятости, если эти биты будут присвоены в будущем.

Н. Для пакетов пересылки битовых блоков

Поля Значения верхней левой координаты X окна и Значения верхней левой координаты Y окна используют каждое 2 байта для задания значения X и Y координат верхнего левого
35 угла окна, подлежащего перемещению. Поля Ширины и Высоты окна используют каждое 2 байта для задания ширины и высоты окна, подлежащего перемещению. Поля Перемещение окна по оси X и Перемещение окна по оси Y используют каждое 2 байта для задания количества пикселей, на которое окно должно быть перемещено в горизонтальном и вертикальном направлении соответственно. Обычно эти координаты конфигурируются
40 так, что положительные значения для оси X вызывают перемещение окна вправо, а отрицательные значения вызывают перемещение влево, тогда как положительные значения для оси Y вызывают перемещение окна вниз, а отрицательные значения вызывают перемещение вверх.

И. Для пакетов заливки растровой области

45 Поля Значения верхней левой координаты X окна и Значения верхней левой координаты Y окна используют каждое 2 байта для задания значения X и Y координат верхнего левого угла окна, подлежащего заливке. Поля Ширины окна и Высоты окна (каждое 2 байта) задают ширину и высоту окна, подлежащего заливке. Поле Дескриптора формата видеоданных (2 байта) задает формат Значения пикселя заливки области. Формат
50 аналогичен формату аналогичного поля в Пакете видеопотока. Поле (4 байта) Значения пикселя заливки области содержит значение пикселя, подлежащего заливке в окно, задаваемое описанными выше полями. Формат этого пикселя задается в поле Дескриптора формата видеоданных.

J. Для пакетов заливки растровым узором

Поля Значения верхней левой координаты X окна и Значения верхней левой координаты Y окна каждое использует 2 байта для задания значения X и Y координат верхнего левого угла окна, подлежащего заливке. Поля Ширины окна и Высоты окна (каждое 2 байта)

- 5 задают ширину и высоту окна, подлежащего заливке. Поля (каждое 2 байта) Ширины узора и Высоты узора задают ширину и высоту соответственно узора заливки. Поле (2 байта) Смещения узора по горизонтали задает смещение по горизонтали узора данных пикселей от левого края заданного окна, подлежащего заливке. Задаваемое значение должно быть меньше, чем значение в Поле ширины узора. Поле (2 байта) Смещения узора по вертикали
- 10 задает смещение по вертикали узора данных пикселей от верхнего края заданного окна, подлежащего заливке. Задаваемое значение должно быть меньше, чем значение в поле Высоты узора.

2-байтовое поле Дескриптора формата видеоданных задает формат Значения пикселя заливки области. Фиг.11 иллюстрирует, как кодируется Дескриптор формата видеоданных.

- 15 Формат аналогичен формату аналогичного поля в Пакете видеопотока.

Поле (2 байта) ЦИК параметров содержит ЦИК всех байтов от Длины пакета до Дескриптора формата видео. Если проверка этим ЦИК unsuccessful, тогда весь пакет отбрасывается. Поле Данных пикселей узора содержит необработанную видеоинформацию, которая задает узор заливки в формате, задаваемым Дескриптором

20 формата видеоданных. Данные упакованы в байты, и первый пиксель каждой строки должен быть выровнен по байтам. Данные узора заливки передаются построчно. Поле (2 байта) ЦИК данных пикселей узора содержит ЦИК только Данных пикселей узора. Если проверка этим ЦИК unsuccessful, тогда Данные пикселей узора все же могут использоваться, но число ошибок ЦИК получает приращение.

- 25 **К. Пакеты канала данных линии обмена данными**

Поле (2 байта) ЦИК параметров содержит 16-битовый ЦИК всех байтов с Длины пакета до Типа пакета. Если проверка этим ЦИК unsuccessful, тогда весь пакет отбрасывается.

Поле Данных линии обмена данными содержит необработанные данные из канала обмена данными. Эти данные просто пропускаются на вычислительное устройство в дисплее.

- 30 Поле (2 байта) ЦИК данных линии обмена данными содержит 16-битовый ЦИК только для Данных линии обмена данными. Если проверка этим ЦИК unsuccessful, тогда Данные линии обмена данными все же используются или являются полезными, но число ошибок ЦИК получает приращение.

- 35 **L. Для пакетов включения прямых аудиоканалов**

Поле (1 байт) Маски включения аудиоканалов содержит группу флагов, которые указывают, какие аудиоканалы должны быть включены в клиенте. Бит, установленный в единицу, включает соответствующий канал, и бит, установленный в нуль, выключает соответствующий канал. Биты 0-5 обозначают каналы 0-5, которые адресуют левый

40 передний, правый передний, левый задний, правый задний, передний центральный и сверхнизкочастотный каналы соответственно. Биты 6 и 7 резервируются для будущего использования и в настоящее время, как правило, устанавливаются равными нулю.

M. Для пакетов частоты обратных аудиоотсчетов

Поле (1 байт) Частоты аудиоотсчетов задает частоту цифровых аудиоотсчетов.

- 45 Значения для этого поля присваиваются различным частотам, при этом значения 0, 1, 2, 3, 4, 5, 6, 7 и 8 используются для обозначения 8000, 16000, 24000, 32000, 40000, 48000, 11025, 22050 и 44100 отсчетов в секунду (отсчет/с) соответственно, причем значения 9-254 резервируются для использования с альтернативными частотами, когда потребуется, поэтому они в настоящее время устанавливаются в «0». Значение 255
- 50 используется для выключения аудиопотока обратной линии передачи данных.

Поле (1 байт) Формата отсчета задает формат цифровых аудиоотсчетов. Когда Биты [1:0] равны «0», то цифровые аудиоотсчеты находятся в линейном формате, когда они равны 1, то цифровые аудиоотсчеты находятся в формате по закону «мю», а когда они равны 2,

то цифровые аудиоотсчеты находятся в формате по А-закону. Биты [7:2] резервируются для альтернативного использования при обозначении форматов аудио, когда потребуется, и, как правило, устанавливаются равными нулю.

Н. Для пакетов служебных данных защиты цифрового содержимого

- 5 Поле (1 байт) Типа защиты содержимого задает способ защиты цифрового содержимого, который используется. Значение «0» указывает защиту цифрового содержимого передачи (ЗЦСП), тогда как значение 1 указывает широкополосную систему защиты цифрового содержимого (ШСЗЦС). Диапазон значений 2-255 в настоящее время не задается, но резервируется для использования с альтернативными схемами защиты, когда потребуется.
- 10 Поле Служебных сообщений защиты содержимого представляет собой поле с переменной длиной, содержащее сообщения защиты содержимого, пересылаемые между хостом и клиентом.

О. Для пакетов разрешения прозрачного цвета

- Поле (1 байт) Разрешения прозрачного цвета задает, когда разрешается или
- 15 запрещается режим прозрачного цвета. Если Бит 0 равен нулю, то тогда запрещается режим прозрачного цвета, если он равен 1, то тогда разрешается режим прозрачного цвета и прозрачный цвет задается следующими двумя параметрами. Биты 1-7 этого байта резервируются для будущего использования и обычно устанавливаются равными нулю.

- Поле (2 байта) Дескриптора формата видеоданных задает формат Значения пикселя заливки области. Фиг.11 иллюстрирует, как кодируется Дескриптор формата видеоданных. Формат в основном аналогичен формату этого же поля в Пакете Видео потока.
- 20

Поле Значения пикселя заливки области использует 4 байта, выделенные для значения пикселя, подлежащего заливке в заданное выше окно. Формат этого пикселя задается в поле Дескриптора формата видеоданных.

- 25 **Р. Для пакетов измерения задержки на двойное прохождение**

- 2-байтовое поле Длины пакета задает общее количество байтов в пакете, не включая поле длины пакета, и в одном варианте осуществления выбирается таким, что имеет фиксированную длину 159. 2-байтовое поле Типа пакета, которое идентифицирует этот тип пакета со значением 82, идентифицирует пакет в качестве Пакета измерения задержки на
- 30 двойное прохождение. Поле ИД hClient, как и раньше, резервируется для будущего использования в качестве ИД клиента и, как правило, устанавливается в нуль.

В одном варианте осуществления поле (2 байта) ЦИК параметров содержит 16-битовый ЦИК всех байтов с Длины пакета до Типа пакета. Если проверка этим ЦИК не успешна, тогда весь пакет отбрасывается.

- 35 Поле (в данном случае 64 байта) Защитного интервала 1 используется для того, чтобы дать возможность линейным драйверам MDDI_Data в клиенте включиться до выключения линейных драйверов в хосте. Клиент включает свои линейные драйверы MDDI_Data в течение бита 0 Защитного интервала 1, и хост выключает свои линейные драйверы, чтобы они были полностью выключены до последнего бита Защитного интервала 1. Хост и клиент
- 40 оба возбуждают уровнем логического нуля в течение Защитного интервала 1, когда они не выключены. Другим назначением этого поля является то, чтобы обеспечивать, чтобы все сигналы MDDI_Data находились на уровне логического нуля в течение достаточного времени, чтобы дать возможность клиенту начать восстановление тактовых импульсов или тактового сигнала, используя только MDDI_Stb, до выключения линейных драйверов хоста.

- 45 Поле Периода измерения представляет собой 64-байтовое окно, используемое для того, чтобы дать возможность клиенту ответить двумя байтами 0xff и 30 байтами 0x00 с половинной скоростью передачи данных, используемой на прямой линии передачи данных. Эта скорость передачи данных соответствует Делителю скорости передачи обратной линии передачи данных, равному 1. Клиент возвращает этот ответ немедленно в момент, когда
- 50 он воспринимает начало Периода измерения. Этот ответ от клиента будет принят хостом точно с задержкой на двойное прохождение линии передачи данных плюс логическая задержка в клиенте после начала первого бита Периода измерения на хосте.

Поле (2 байта) Всех нулей 1 содержит нули, чтобы дать возможность линейным

драйверам MDDI_Data в хосте и клиенте выполнить перекрытие, так что MDDI_Data всегда возбуждается. Хост включает линейные драйверы MDDI_Data в течение бита 0 поля Всех нулей 1, и клиент также продолжает возбуждать сигнал уровнем логического нуля, как он делал в конце Периода измерения.

- 5 Значение в поле (64 байта) Защитного интервала 2 позволяет получить перекрытие Периода измерения, возбуждаемого клиентом, когда задержка на двойное прохождение имеет максимальную величину, которая может быть измерена в Периоде измерения. Клиент выключает свои линейные драйверы в течение бита 0 Защитного интервала 2, и Хост включает свои линейные драйверы непосредственно после последнего бита
- 10 Защитного интервала 2. Хост и клиент оба возбуждают уровнем логического нуля в течение Защитного интервала 2, когда они не выключены. Другим назначением этого поля является обеспечение того, чтобы все сигналы MDDI_Data находились на уровне логического нуля в течение достаточного времени, чтобы дать возможность клиенту начать восстановление тактового сигнала, используя как MDDI_Data0, так и MDDI_Stb
- 15 после включения линейных драйверов для хоста.

Q. Для пакетов калибровки перекося прямой линии передачи данных

В одном варианте осуществления поле (2 байта) ЦИК параметров содержит 16-битовый ЦИК всех байтов с Длины пакета до Типа пакета. Если проверка этим ЦИК неуспешна, тогда весь пакет отбрасывается.

- 20 Поле Всех нулей 1 использует 8 байтов для обеспечения того, что будут переходы по MDDI_Stb в начале поля Последовательности данных калибровки. Как правило, эти байты используют 8-битовые целые числа без знака, равные нулю. Оно также обеспечивает достаточное время, чтобы логика ядра клиента переключила режим схемы восстановления тактовых импульсов с использования исключающего ИЛИ (XOR) MDDI_0 и MDDI_Stb на
- 25 простое использование MDDI_Stb или сигнала MDDI_Stb в качестве восстановленных тактовых импульсов.

- Поле Последовательности данных калибровки содержит последовательность данных, которая вызывает переключение сигналов MDDI_Data при каждом периоде данных. Длина поля Последовательности данных калибровки определяется интерфейсом, используемым
- 30 на прямой линии передачи данных. Во время обработки Последовательности данных калибровки контроллер хоста ЦИМД устанавливает все сигналы MDDI_Data равными строб-сигналу. Схема восстановления тактовых импульсов клиента должна использовать только MDDI_Stb, а не MDDI_Stb XOR MDDI_Data0 для восстановления тактовых импульсов данных, когда поле Последовательности данных калибровки принимается
- 35 клиентом. В зависимости от точной фазы сигнала MDDI_Stb в начале поля Последовательности данных калибровки Последовательность данных калибровки, как правило, будет одной из следующих, основываясь на Типе интерфейса, используемом тогда, когда посылается этот пакет:

- 40 Тип 1 - (64-байтовая последовательность данных) 0хаа, 0хаа ... или 0х55, 0х55 ...
 Тип 2 - (128-байтовая последовательность данных) 0хсс, 0хсс ... или 0х33 0х33 ...
 Тип 3 - (256-байтовая последовательность данных) 0xf0, 0xf0 ... или 0x0f, 0x0f ...
 Тип 4 - (512-байтовая последовательность данных) 0xff, 0x00, 0xff, 0x00 ... или 0x00, 0xff, 0x00, 0xff ...

- Поле Всех нулей 2 использует 8 байтов для обеспечения достаточного времени, чтобы
- 45 логика ядра клиента переключила режим схемы восстановления тактовых импульсов обратно в исходное состояние из использования сигнала MDDI_Stb в качестве восстановленного тактового сигнала в использование исключающего ИЛИ сигналов MDDI_0 и MDDI_Stb. Как правило, эти байты используют 8-битовые целые числа без знака, равные нулю.

- 50 Примеры возможных форм сигнала MDDI_Data и MDDI_Stb для обоих Интерфейсов Типа 1 и Типа 2 показаны на фиг.62А и 62В соответственно.

XX. Заключение

Хотя выше были описаны различные варианты осуществления настоящего изобретения,

должно быть понятно, что они были представлены только в качестве примера, а не ограничения. Таким образом, широта и объем настоящего изобретения не должны ограничиваться никаким из вышеописанных примерных вариантов осуществления, но должны определяться только согласно следующей формуле изобретения и ее

5 эквивалентам.

Формула изобретения

1. Способ подтверждения того, что клиент находится в синхронизме с хостом, и выполнения динамической смены с первого количества каналов передачи данных на

10 второе количество каналов передачи данных в линии цифровой передачи, содержащий этапы, на которых

посылают посредством хоста на клиент пакет инкапсуляции обратной линии передачи данных,

посылают посредством клиента на хост пакет ответа, подтверждают посредством хоста,

15 что пакет ответа был корректно принят на хосте, и

предписывают клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных.
2. Способ по п.1, в котором на этапе отправки пакета ответа от клиента на хост посылают пакет запроса клиента и состояния.
- 20 3. Способ по п.1, в котором переход содержит независимый переход прямой линии связи и независимый переход обратной линии связи.
4. Способ по п.1, дополнительно содержащий этап, на котором предоставляют заранее определенное количество времени для подготовки клиента к переходу.
5. Система для подтверждения того, что клиент находится в синхронизме с хостом, и

25 выполнения динамической смены с первого количества каналов передачи данных на второе количество каналов передачи данных в линии цифровой передачи, содержащая средство для отправки посредством хоста на клиент пакета инкапсуляции обратной линии передачи данных,

средство для отправки посредством клиента на хост пакета ответа,

30 средство для подтверждения посредством хоста, что пакет ответа был корректно принят на хосте, и

средство для предписания клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных.
- 35 6. Система по п.5, в которой средство для отправки пакета ответа от клиента на хост содержит средство для отправки пакета запроса клиента и состояния.
7. Система по п.5, в которой переход содержит независимый переход прямой линии связи и независимый переход обратной линии связи.
8. Система по п.5, дополнительно содержащая средство для предоставления заранее

40 определенного количества времени для подготовки клиента к переходу.
9. Машиночитаемый носитель, содержащий компьютерный код для предписания выполнения подтверждения того, что клиент находится в синхронизме с хостом, и выполнения динамической смены с первого количества каналов передачи данных на второе количество каналов передачи данных в линии цифровой передачи, причем

45 упомянутый компьютерный код содержит

код для предписания выполнения хостом отправки на клиент пакета инкапсуляции обратной линии передачи данных,

код для предписания выполнения клиентом отправки на хост пакета ответа,

код для предписания выполнения хостом подтверждения того, что пакет ответа был

50 корректно принят на хосте, и

код для предписания клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных.
10. Машиночитаемый носитель по п.9, в котором код для предписания выполнения

посылки пакета ответа от клиента на хост содержит код для предписания выполнения посылки пакета запроса клиента и состояния.

11. Машиночитаемый носитель по п.9, в котором переход содержит независимый переход прямой линии связи и независимый переход обратной линии связи.

5 12. Машиночитаемый носитель по п.9, в котором компьютерный код дополнительно содержит код для предписания выполнения предоставления заранее определенного количества времени для подготовки клиента к переходу.

13. Способ выполнения динамической смены с первого количества каналов передачи данных на второе количество каналов передачи данных в линии цифровой передачи, содержащий этапы, на которых

10 посылают на клиент пакет выполнения перехода на тип, предписывающий клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных, и

15 принимают посредством клиента второе количество каналов передачи данных при приеме пакета, следующего за посланным пакетом выполнения перехода на тип.

14. Способ по п.13, дополнительно содержащий этапы, на которых посылают посредством хоста на клиент пакет инкапсуляции обратной линии передачи данных,

20 посылают посредством клиента на хост пакет ответа, подтверждают посредством хоста, что пакет ответа был корректно принят.

15. Способ по п.14, в котором на этапе посылки пакета ответа от клиента на хост посылают пакет запроса клиента и состояния.

25 16. Способ по п.13, в котором пакет выполнения перехода на тип содержит пакет независимого перехода прямой линии связи и пакет независимого перехода обратной линии связи.

17. Способ по п.13, дополнительно содержащий этап, на котором предоставляют заранее определенное количество времени для подготовки клиента к приему второго количества каналов передачи данных.

30 18. Система для выполнения динамической смены с первого количества каналов передачи данных на второе количество каналов передачи данных в линии цифровой передачи, содержащая

средство для посылки на клиент пакета выполнения перехода на тип, предписывающего клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных, и

35 средство для приема посредством клиента второго количества каналов передачи данных при приеме пакета, следующего за посланным пакетом выполнения перехода на тип.

40 19. Система по п.18, дополнительно содержащая средство для посылки посредством хоста на клиент пакета инкапсуляции обратной линии передачи данных,

средство для посылки посредством клиента на хост пакета ответа,

средство для подтверждения посредством хоста того, что пакет ответа был корректно принят.

45 20. Система по п.19, в которой средство для посылки пакета ответа от клиента на хост содержит средство для посылки пакета запроса клиента и состояния.

21. Система по п.18, в которой пакет выполнения перехода на тип содержит пакет независимого перехода прямой линии связи и пакет независимого перехода обратной линии связи.

50 22. Система по п.18, дополнительно содержащая средство для предоставления заранее определенного количества времени для подготовки клиента к приему второго количества каналов передачи данных.

23. Машиночитаемый носитель, содержащий компьютерный код для предписания выполнения динамической смены с первого количества каналов передачи данных на

второе количество каналов передачи данных в линии цифровой передачи, причем компьютерный код содержит

код для предписания выполнения посылки на клиент пакета выполнения перехода на тип, предписывающего клиенту выполнить переход от использования первого количества каналов передачи данных к использованию второго количества каналов передачи данных, и

код для предписания выполнения клиентом приема второго количества каналов передачи данных при приеме пакета, следующего за посланным пакетом выполнения перехода на тип.

24. Машиночитаемый носитель по п.23, в котором компьютерный код дополнительно содержит

код для предписания выполнения хостом посылки на клиент пакета инкапсуляции обратной линии передачи данных,

код для предписания выполнения клиентом посылки на хост пакета ответа,

код для предписания выполнения хостом подтверждения того, что пакет ответа был корректно принят.

25. Машиночитаемый носитель по п.24, в котором код для предписания выполнения посылки пакета ответа от клиента на хост содержит код для предписания выполнения посылки пакета запроса клиента и состояния.

26. Машиночитаемый носитель по п.23, в котором пакет выполнения перехода на тип содержит пакет независимого перехода прямой линии связи и пакет независимого перехода обратной линии связи.

27. Машиночитаемый носитель по п.23, в котором компьютерный код дополнительно содержит код для предписания выполнения предоставления заранее определенного количества времени для подготовки клиента к приему второго количества каналов передачи данных.

28. Способ модифицирования атрибутов дисплея в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), содержащий этапы, на которых посылают от хоста на дисплей пакет состояния питания дисплея, содержащий

модифицированные атрибуты дисплея,

подтверждают посредством клиента прием пакета состояния питания дисплея и реализуют посредством клиента модифицированные атрибуты дисплея.

29. Способ по п.28, в котором атрибуты дисплея содержат состояние питания дисплея и состояние питания подсветки.

30. Система для модифицирования атрибутов дисплея в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), содержащая

средство для посылки от хоста на дисплей пакета состояния питания дисплея,

содержащего модифицированные атрибуты дисплея,

средство для подтверждения посредством клиента приема пакета состояния питания дисплея и

средство для реализации посредством клиента модифицированных атрибутов дисплея.

31. Система по п.30, в которой атрибуты дисплея содержат состояние питания дисплея и состояние питания подсветки.

32. Машиночитаемый носитель, содержащий компьютерный код для предписания

выполнения модифицирования атрибутов дисплея в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), при этом компьютерный код содержит

код для предписания выполнения посылки от хоста на дисплей пакета состояния питания дисплея, содержащего модифицированные атрибуты дисплея,

код для предписания выполнения клиентом подтверждения приема пакета состояния питания дисплея и

код для предписания выполнения клиентом реализации модифицированных атрибутов дисплея.

33. Машиночитаемый носитель п.32, в которой атрибуты дисплея содержат состояние

питания дисплея и состояние питания подсветки.

34. Способ направления данных пикселей из пакета масштабируемого видеопотока в клиенте в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), содержащий этапы, на которых

- 5 принимают посредством клиента от хоста содержимое заранее заданных пикселей, масштабируют посредством клиента упомянутое содержимое заранее заданных пикселей до конкретного размера,
- посылают масштабированное содержимое пикселей, в, по меньшей мере, один конкретный буфер дисплея из множества буферов дисплея, задаваемый полем атрибутов
- 10 данных пикселей, принятым клиентом, и
- отображают содержимое пикселей из упомянутого, по меньшей мере, одного конкретного буфера дисплея.

- 15 35. Способ по п.34, в котором упомянутый, по меньшей мере, один конкретный буфер дисплея включает в себя изображение, отображаемое для левого глаза или для правого глаза.

36. Способ по п.34, дополнительно содержащий этап, на котором считывают посредством контроллера дисплея упомянутый, по меньшей мере, один конкретный буфер дисплея для обновления изображения, отображаемого в текущий момент.

- 20 37. Способ по п.34, в котором упомянутый, по меньшей мере, один конкретный буфер дисплея содержит автономный буфер изображения.

38. Система для направления данных пикселей из пакета масштабируемого видеопотока в клиенте в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), содержащая

- 25 средство для приема посредством клиента от хоста содержимого заранее заданных пикселей,
- средство для масштабирования посредством клиента упомянутого содержимого заранее заданных пикселей до конкретного размера,
- средство для отправки масштабированного содержимого пикселей в, по меньшей мере, один конкретный буфер дисплея из множества буферов дисплея, задаваемый полем
- 30 атрибутов данных пикселей, принятым клиентом, и
- средство для отображения содержимого пикселей из упомянутого, по меньшей мере, одного конкретного буфера дисплея.

- 35 39. Система по п.38, в которой упомянутый, по меньшей мере, один конкретный буфер дисплея включает в себя изображение, отображаемое для левого глаза или для правого глаза.

40. Система по п.38, дополнительно содержащая средство для считывания посредством контроллера дисплея упомянутого, по меньшей мере, одного конкретного буфера дисплея для обновления изображения, отображаемого в текущий момент.

- 40 41. Система по п.34, в которой упомянутый, по меньшей мере, один конкретный буфер дисплея содержит автономный буфер изображения.

42. Машиночитаемый носитель, содержащий компьютерный код для предписания выполнения направления данных пикселей из пакета масштабируемого видеопотока в клиенте в системе связи с цифровым интерфейсом для мобильных дисплеев (ЦИМД), при этом компьютерный код содержит

- 45 код для предписания выполнения клиентом приема от хоста содержимого заранее заданных пикселей,
- код для предписания выполнения клиентом масштабирования упомянутого содержимого заранее заданных пикселей до конкретного размера,
- код для предписания выполнения отправки масштабированного содержимого пикселей в,
- 50 по меньшей мере, один конкретный буфер дисплея из множества буферов дисплея, задаваемый полем атрибутов данных пикселей, принятым клиентом, и
- код для предписания выполнения отображения содержимого пикселей из упомянутого, по меньшей мере, одного конкретного буфера дисплея.

43. Машиночитаемый носитель по п.42, в котором упомянутый, по меньшей мере, один конкретный буфер дисплея включает в себя изображение, отображаемое для левого глаза или для правого глаза.

5 44. Машиночитаемый носитель по п.42, в котором компьютерный код дополнительно содержит код для предписания выполнения контроллером дисплея считывания упомянутого, по меньшей мере, одного конкретного буфера дисплея для обновления изображения, отображаемого в текущий момент.

45. Машиночитаемый носитель п.42, в котором упомянутый, по меньшей мере, один конкретный буфер дисплея содержит автономный буфер изображения.

10

15

20

25

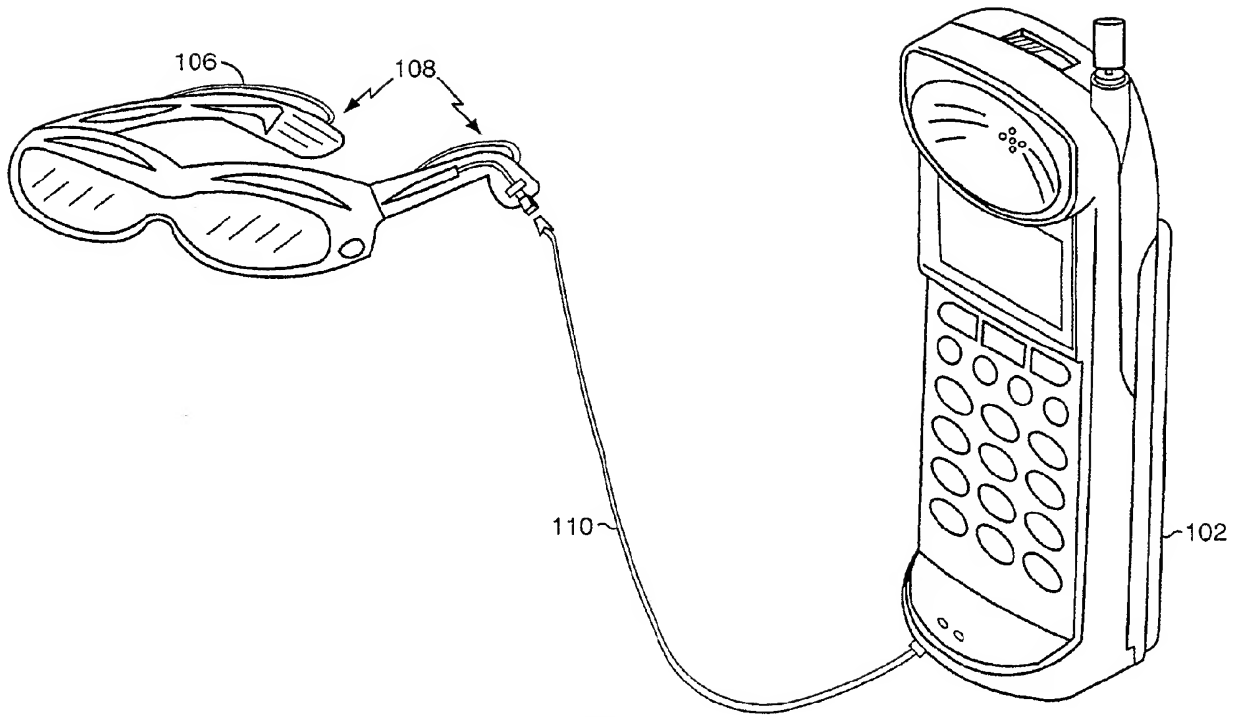
30

35

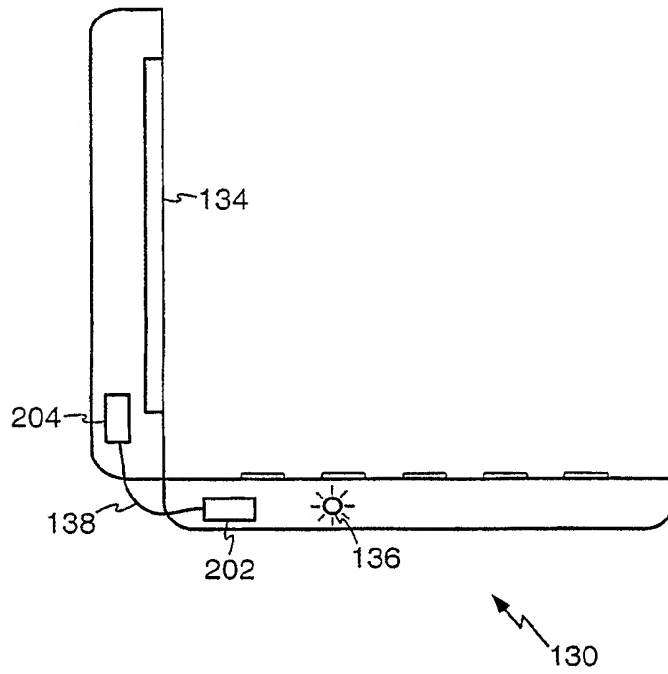
40

45

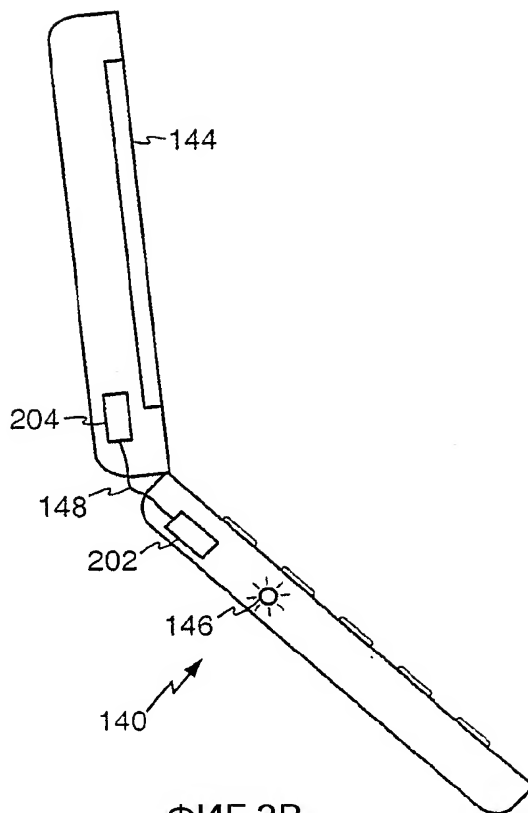
50



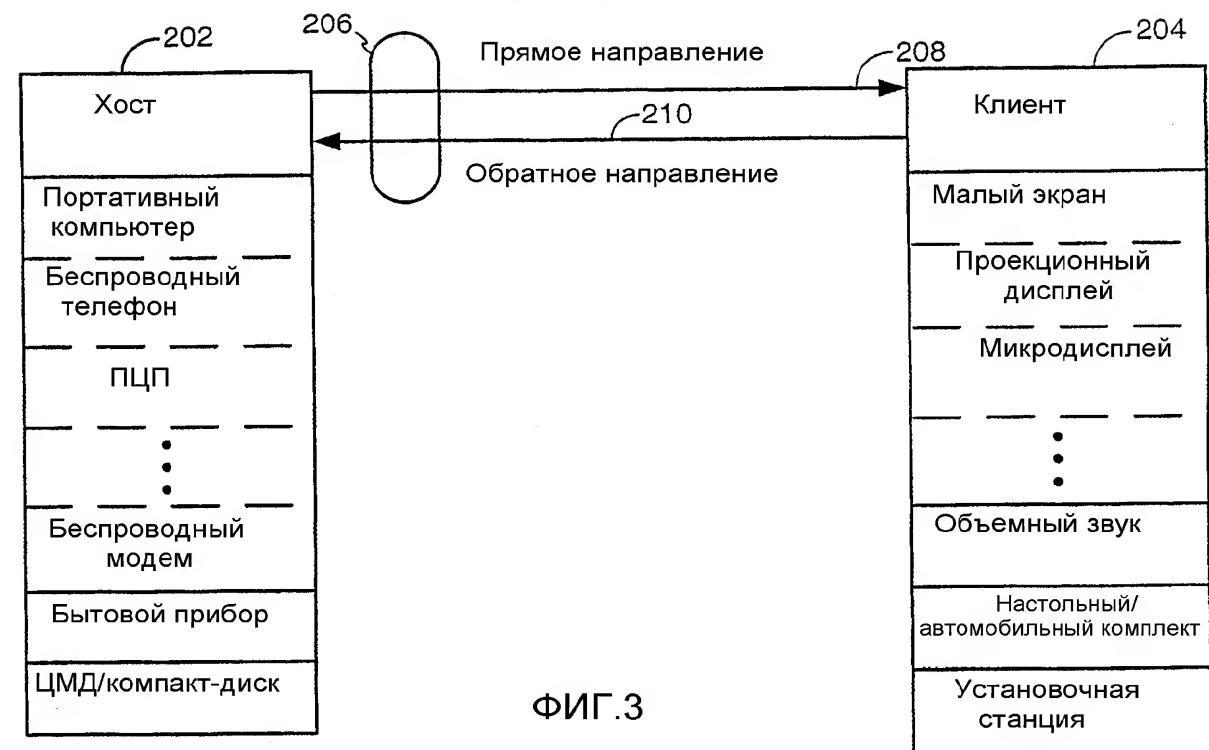
ФИГ.1В



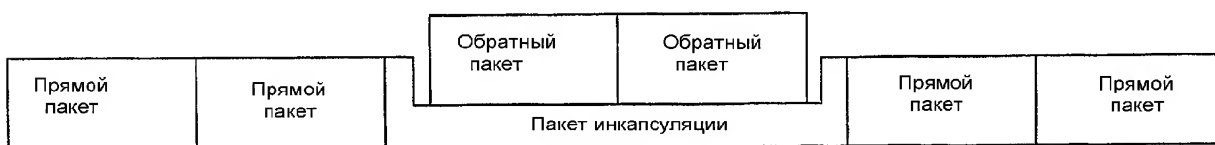
ФИГ.2А



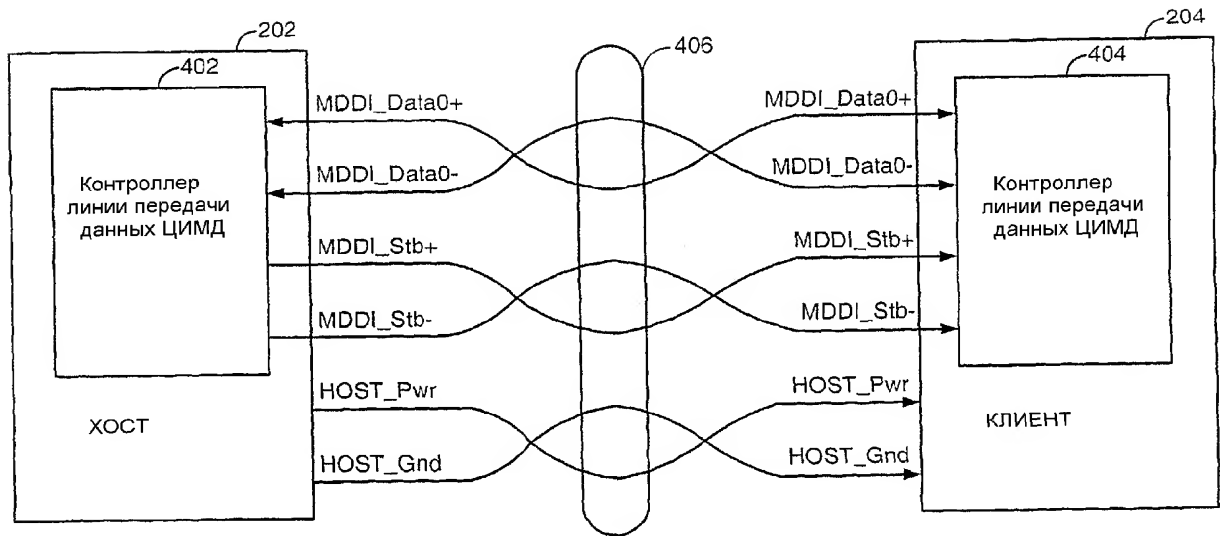
ФИГ.2В



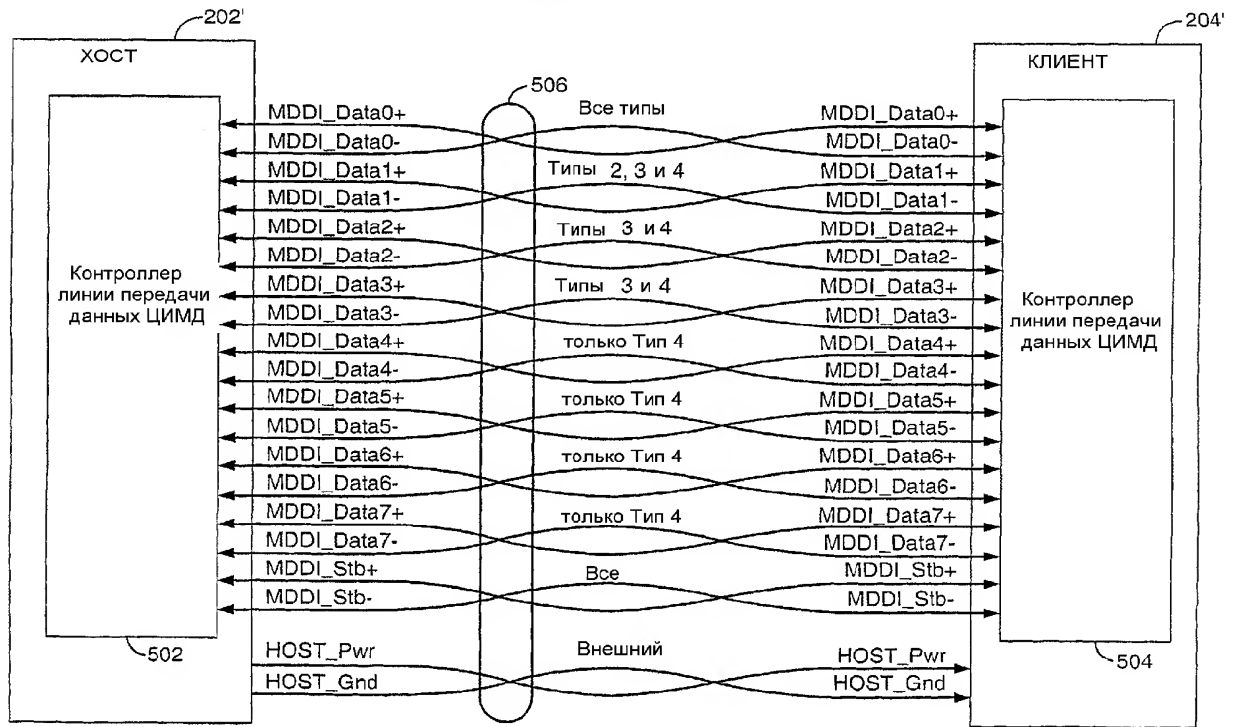
ФИГ.3



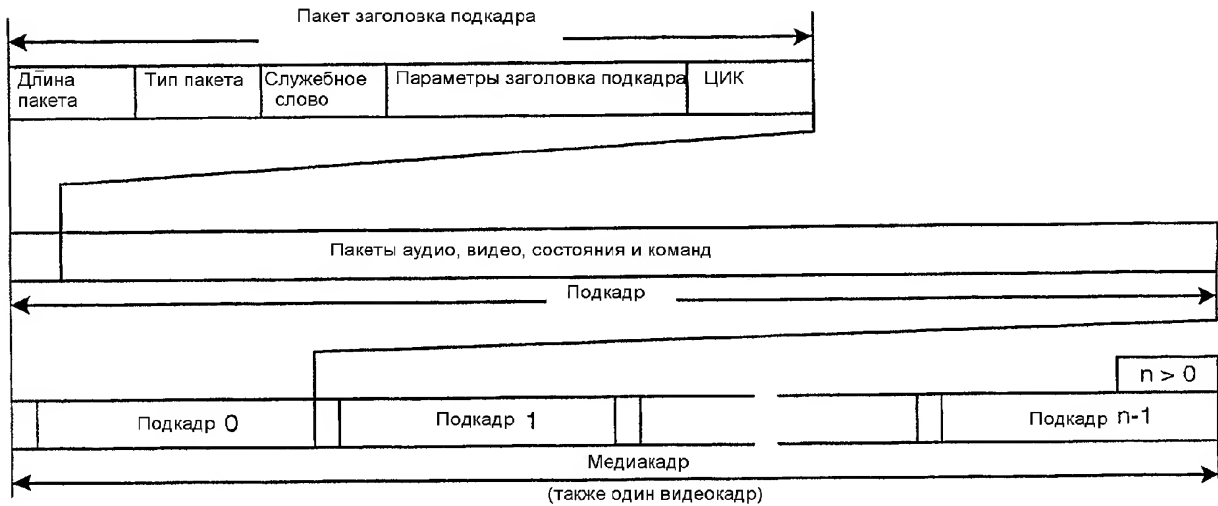
ФИГ.4



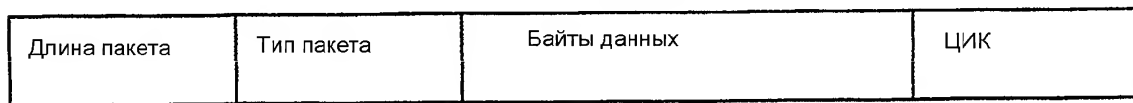
ФИГ. 5



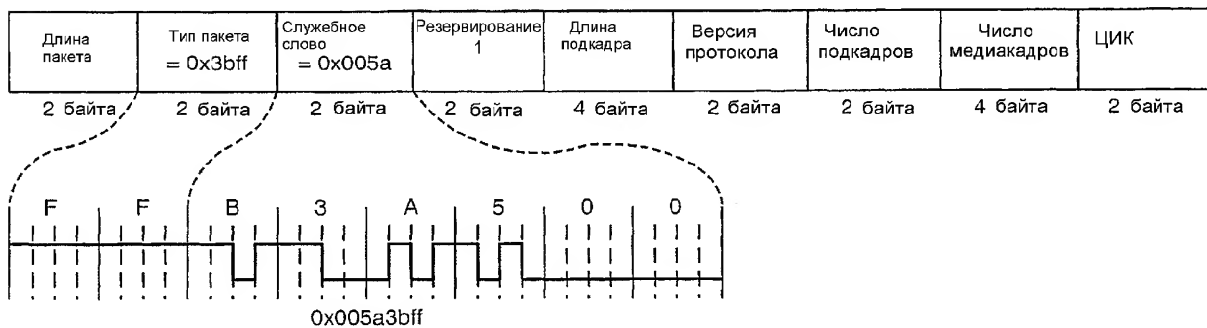
ФИГ. 6



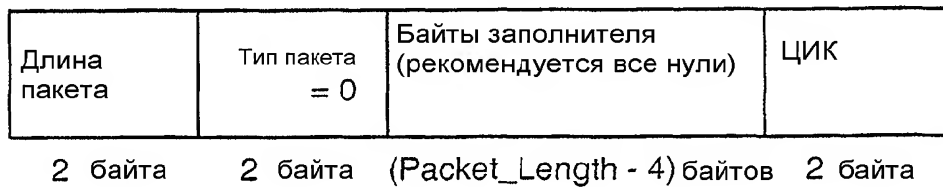
ФИГ.7



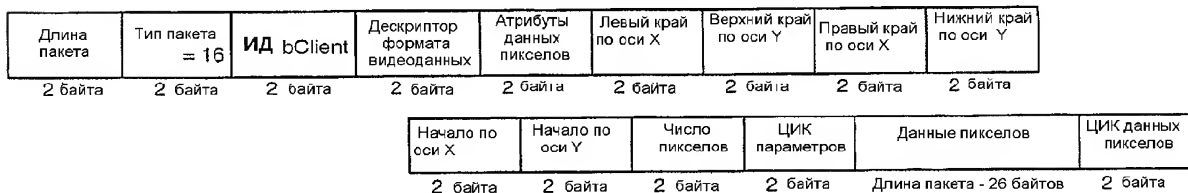
ФИГ.8



ФИГ.9

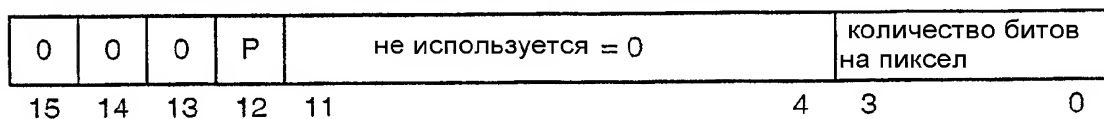


ФИГ.10



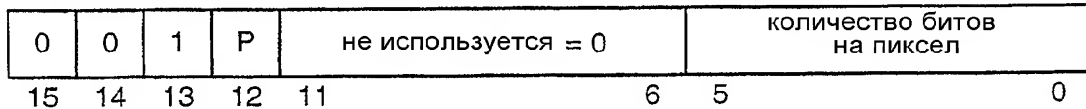
ФИГ.11

Дескриптор формата видеоданных - монохромный



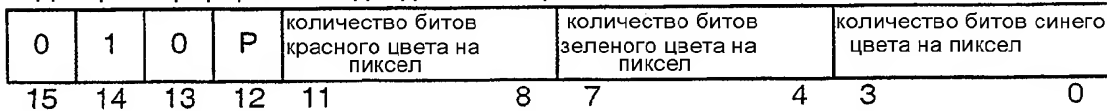
ФИГ.12А

Дескриптор формата видеоданных - цветной с использованием карты цветов



ФИГ.12В

Дескриптор формата видеоданных - цветной, необработанные RGB



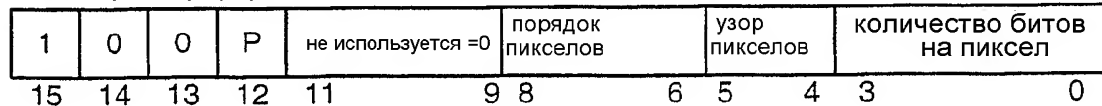
ФИГ.12С

Дескриптор формата видеоданных - видео с цветовыми составляющими, Y, Cb, Cr



ФИГ.12D

Дескриптор формата видеоданных - байеровский



ФИГ.12Е

Выровненный по байтам пиксел формата RGB (не упакован), P=0



Упакованные данные пикселей формата RGB, P=1



Выровненные по байтам пиксели Y Cr Cb (не упакованные), P=0



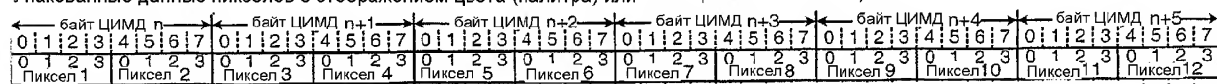
Упакованные данные пикселей Y Cr Cb, P=1



Выровненный по байтам пиксел с отображенным цветом (палитра) или монохромный пиксел (не упакован), P=0

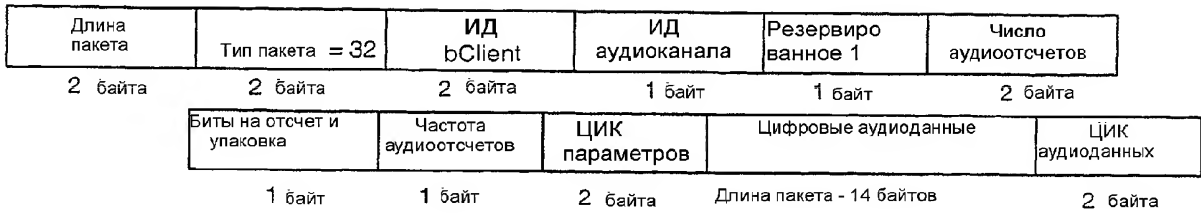


Упакованные данные пикселей с отображением цвета (палитра) или монохромных пикселей, P=1



ФИГ.13

Пакет аудиопотока



ФИГ.14

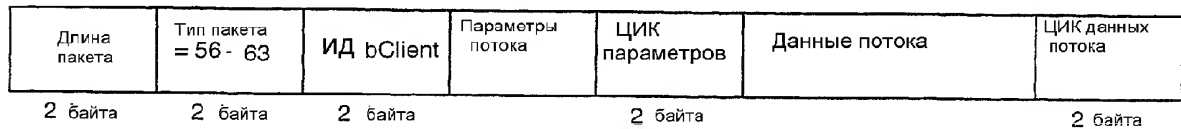
Выверенные по байтам (неупакованные) ИМК-аудиосчеты



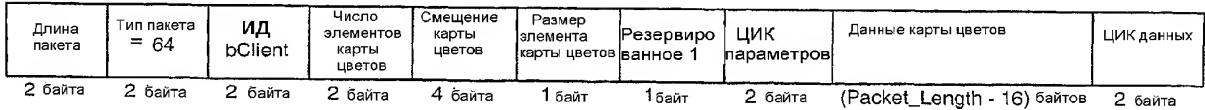
Упакованные ИМК-аудиосчеты



ФИГ.15

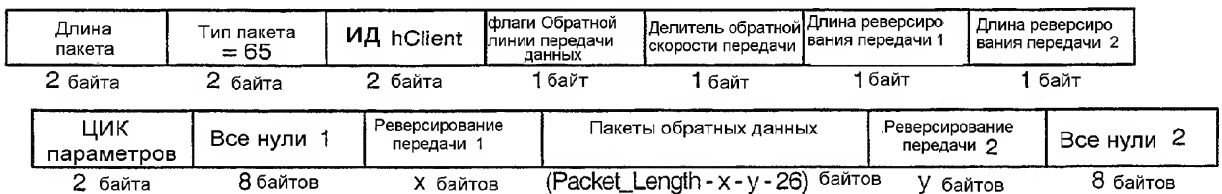


ФИГ.16



ФИГ.17

Пакет инкапсуляции обратной линии передачи данных



ФИГ.18

Пакет возможностей клиента

Длина пакета	Тип пакета = 66	ИД cClient	Версия протокола	Минимальная версия протокола	Возможности скорости передачи данных	Возможности типа интерфейса
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	1 байт
Количество альтернативных дисплеев	Резервированное 1	Ширина растрового изображения	Высота растрового изображения	Ширина окна дисплея	Высота окна дисплея	Размер карты цветов
1 байт	2 байта	2 байта	2 байта	2 байта	2 байта	4 байта
Разрядность карты цветов формата RGB	Возможности формата RGB	Возможности монохромного формата	Резервированное 2	Возможности формата Y Cb Cr	Возможности байерского формата	Резервированное 3
2 байта	2 байта	1 байт	1 байт	2 байта	2 байта	2 байта
Возможности свойств клиента	Максимальная частота видеокадров	Минимальная частота видеокадров	Минимальная частота подкадров	Емкость буфера аудио	Возможности аудиоканала	Возможности частоты аудиосчетов
4 байта	1 байт	1 байт	2 байта	2 байта	2 байта	2 байта
Разрешение аудиосчетов	Разрешение отсчетов микрофона	Возможности частоты отсчетов микрофона	Формат данных клавиатуры	Формат данных указательного устройства	Тип защиты содержимого	Название изготовителя
1 байт	1 байт	2 байта	1 байт	1 байт	2 байта	2 байта
Код продукта	Резервированное 4	Серийный номер	Неделя изготовления	Год изготовления	ЦИК	
2 байта	2 байта	4 байта	1 байт	1 байт	2 байта	

ФИГ.19

Длина пакета	Тип пакета = 67	ИД bClient	Формат данных клавиатуры	Данные клавиатуры	ЦИК
2 байта	2 байта	2 байта	2 байта	(Packet_Length - 8) байтов	2 байта

ФИГ.20

Длина пакета	Тип пакета = 68	ИД bClient	Формат указательного устройства	Данные указательного устройства	ЦИК
2 байта	2 байта	2 байта	2 байта	(Packet_Length - 8) байтов	2 байта

ФИГ.21

Длина пакета	Тип пакета = 69	ЦИК	Все нули
2 байта	2 байта	2 байта	16 байтов

ФИГ.22

Длина пакета	Тип пакета = 70	ИД cClient	Запрос обретной линии передачи данных	Число ошибок ЦИК	Изменение возможностей	флаги Занятости графики	ЦИК
2 байта	2 байта	2 байта	2 байта	1 байт	1 байт	2 байта	2 байта

ФИГ.23

Длина пакета	Тип пакета = 71	ИД hClient	Верхнее левое значение по оси X	Верхнее левое значение по оси Y
2 байта	2 байта	2 байта	2 байта	2 байта

Ширина окна	Высота окна	Перемещение окна по оси X	Перемещение окна по оси Y	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.24

Длина пакета	Тип пакета = 72	ИД hClient	Верхнее левое значение по оси X	Верхнее левое значение по оси Y
--------------	--------------------	------------	---------------------------------	---------------------------------

2 байта 2 байта 2 байта 2 байта 2 байта

Ширина окна	Высота окна	Дескриптор формата данных	Значение пиксела заливки области	ЦИК
-------------	-------------	---------------------------	----------------------------------	-----

2 байта 2 байта 2 байта 4 байта 2 байта

ФИГ.25

Длина пакета	Тип пакета = 73	ИД hClient	Верхнее левое значение по оси X	Верхнее левое значение по оси Y	Ширина окна	Высота окна
--------------	--------------------	------------	---------------------------------	---------------------------------	-------------	-------------

2 байта 2 байта 2 байта 2 байта 2 байта 2 байта 2 байта

Ширина узора	Высота узора	Дескриптор формата данных	ЦИК параметров	Данные пикселей узора	ЦИК данных параметров
--------------	--------------	---------------------------	----------------	-----------------------	-----------------------

2 байта 2 байта 2 байта 2 байта Длина - 22 байта 2 байта

ФИГ.26

Длина пакета	Тип пакета = 74	ИД hClient	Левый край по оси X	Верхний край по оси Y	Правый край по оси X	Нижний край по оси Y	ЦИК
--------------	--------------------	------------	---------------------	-----------------------	----------------------	----------------------	-----

2 байта 2 байта 2 байта 2 байта 2 байта 2 байта 2 байта 2 байта

ФИГ.27

Пакет состояния питания дисплея

Длина пакета	Тип пакета = 75	ИД hClient	Состояние питания	ЦИК
--------------	--------------------	------------	-------------------	-----

2 байта 2 байта 2 байта 2 байта 2 байта

ФИГ.28

Пакет выполнения перехода на тип

Длина пакета	Тип пакета = 77	Тип интерфейса	Резервированное 1	Заполнитель задержки	ЦИК
--------------	--------------------	----------------	-------------------	----------------------	-----

2 байта 2 байта 1 байт 1 байт Длина пакета - 6 байтов 2 байта

ФИГ.29

Длина пакета	Тип пакета = 78	ИД hClient	Маска включения аудиоканалов	ЦИК
--------------	--------------------	------------	------------------------------	-----

2 байта 2 байта 2 байта 2 байта 2 байта

ФИГ.30

Длина пакета	Тип пакета = 79	ИД hClient	Частота аудиоотсчетов	Резервированное 1	ЦИК
--------------	--------------------	------------	-----------------------	-------------------	-----

2 байта 2 байта 2 байта 1 байт 1 байт 2 байта

ФИГ.31

Длина пакета	Тип пакета = 80	ИД bClient	Тип защиты содержимого	Служебные сообщения защиты содержимого	ЦИК
2 байта	2 байта	2 байта	2 байта	Длина пакета - 8 байтов	2 байта

ФИГ.32

Длина пакета	Тип пакета = 81	ИД hClient	Разрешение прозрачного цвета
2 байта	2 байта	2 байта	1 байт

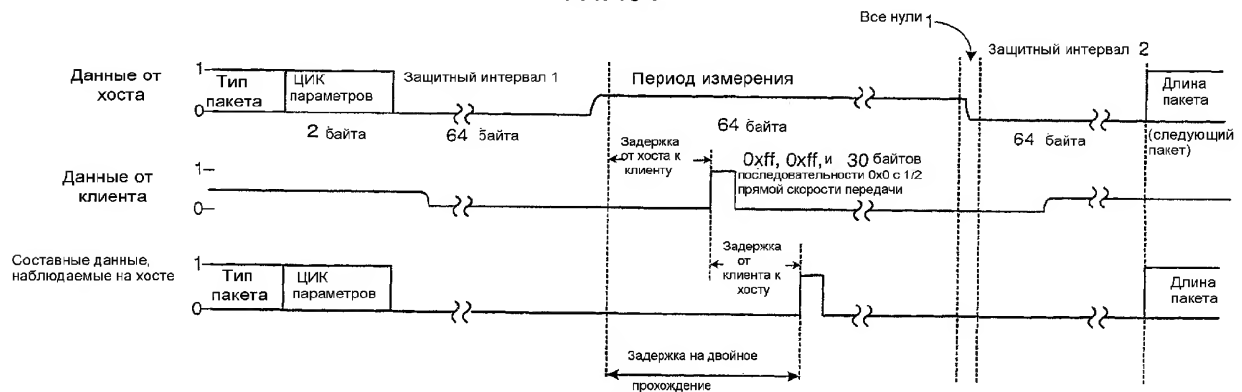
Резервированное 1	Идентификатор альфа-курсора	Дескриптор формата данных	Значение прозрачного пиксела	ЦИК
1 байт	2 байта	2 байта	4 байта	2 байта

ФИГ.33

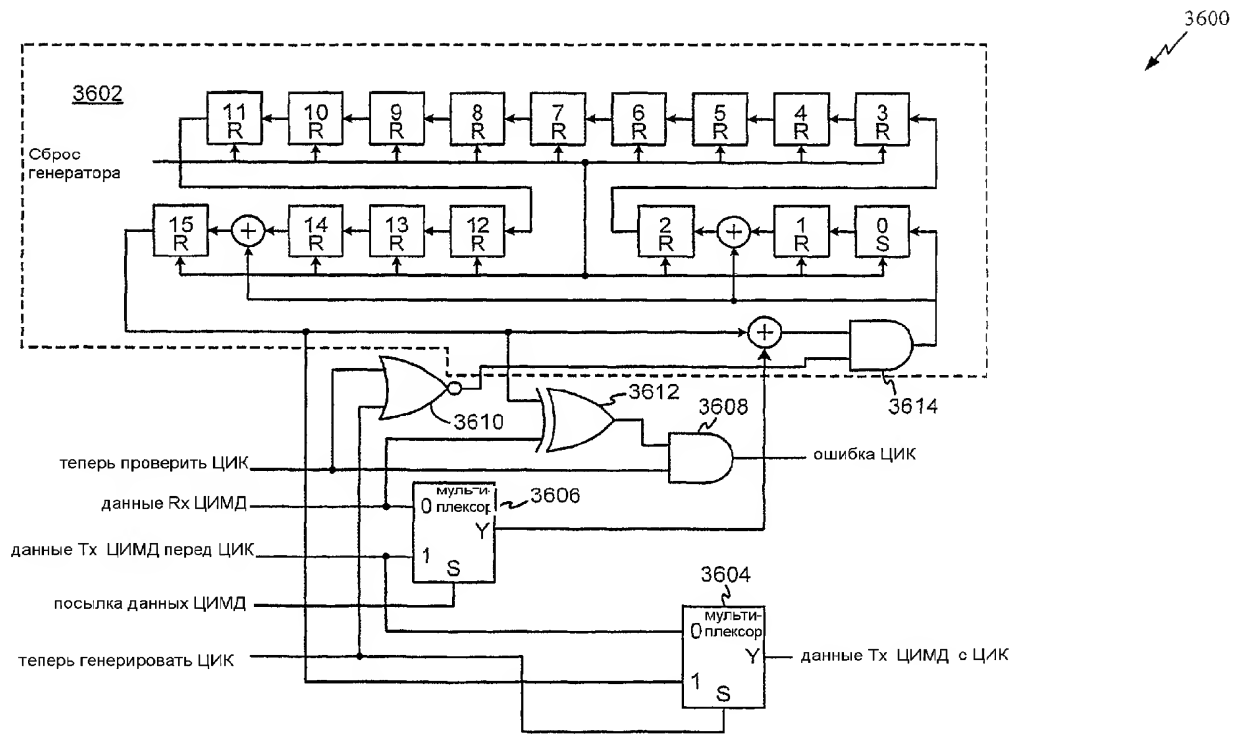
Длина пакета	Тип пакета = 82	ИД hClient	ЦИК параметров
2 байта	2 байта	2 байта	2 байта

Защитный интервал 1	Период измерения	Все нули	Защитный интервал 2
64 байта	64 байта	2 байта	64 байта

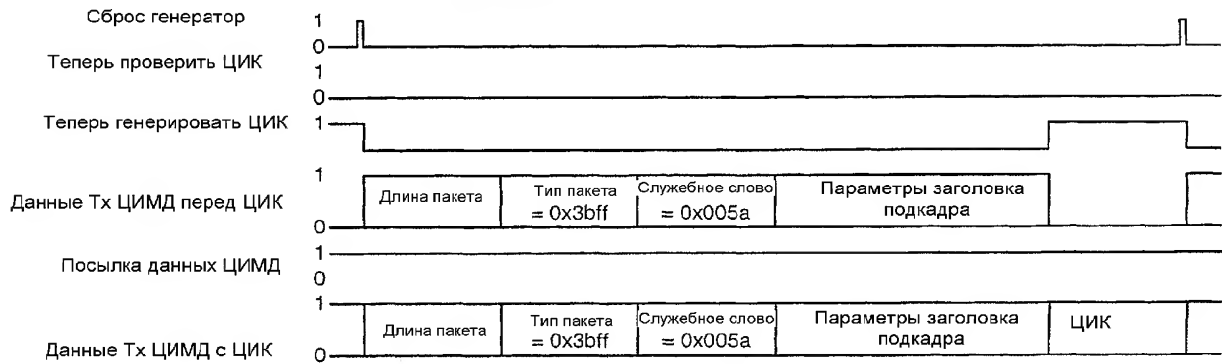
ФИГ.34



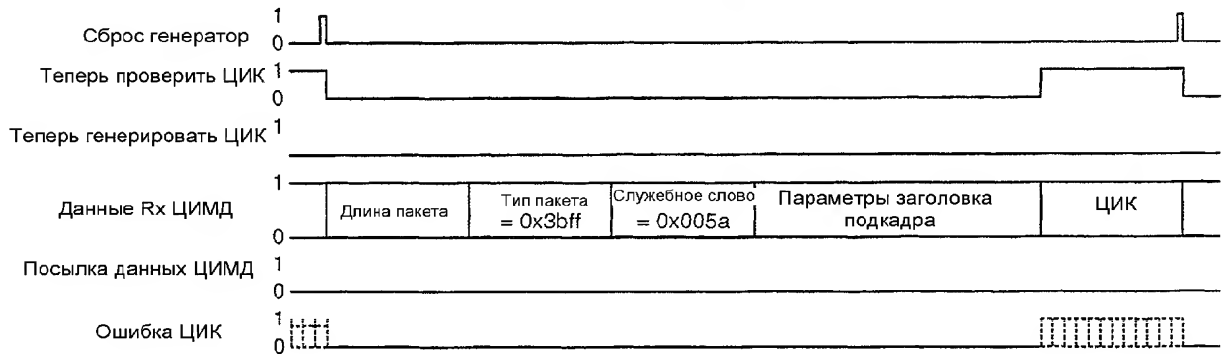
ФИГ.35



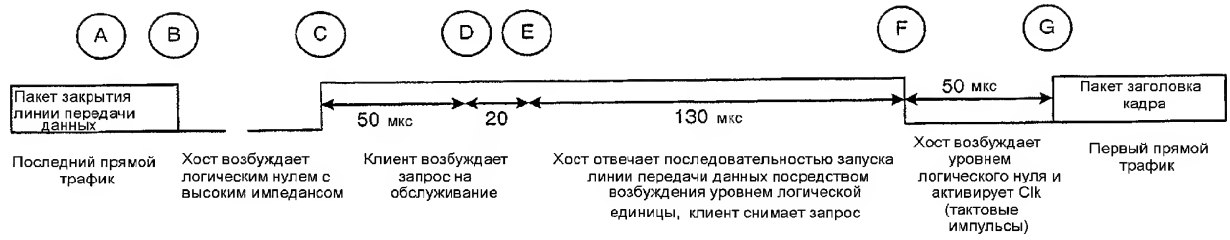
ФИГ.36



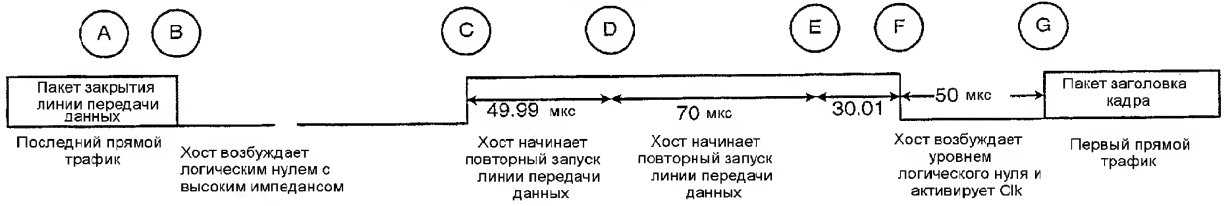
ФИГ.37А



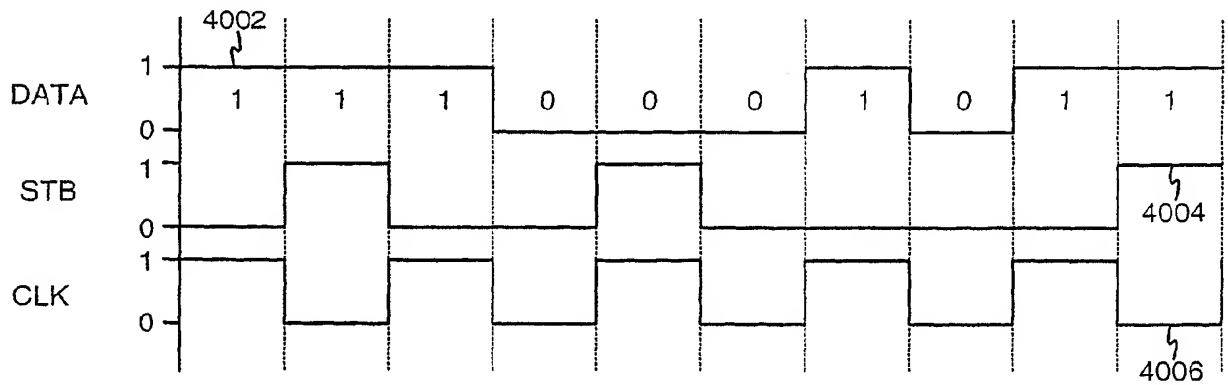
ФИГ.37В



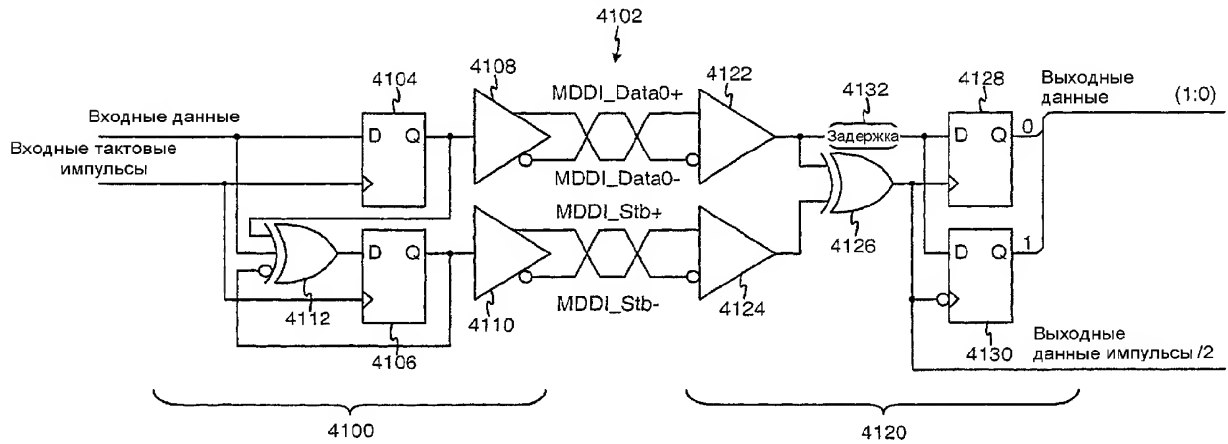
ФИГ.38



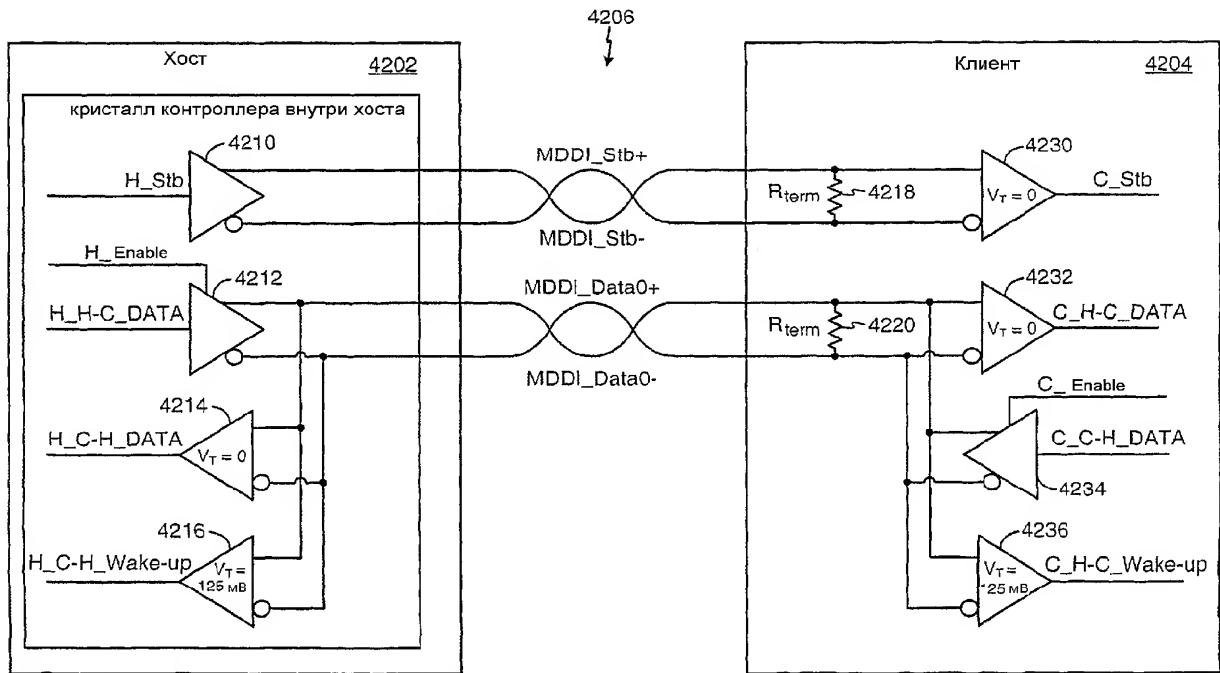
ФИГ.39



ФИГ.40

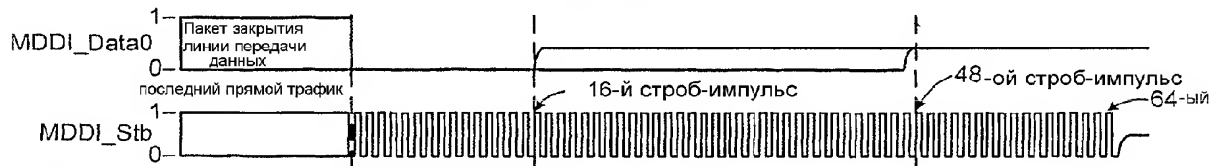


ФИГ.41



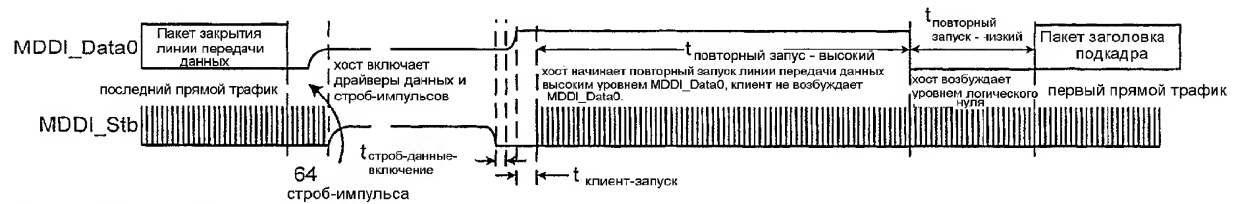
term - согласованной нагрузки

ФИГ.42



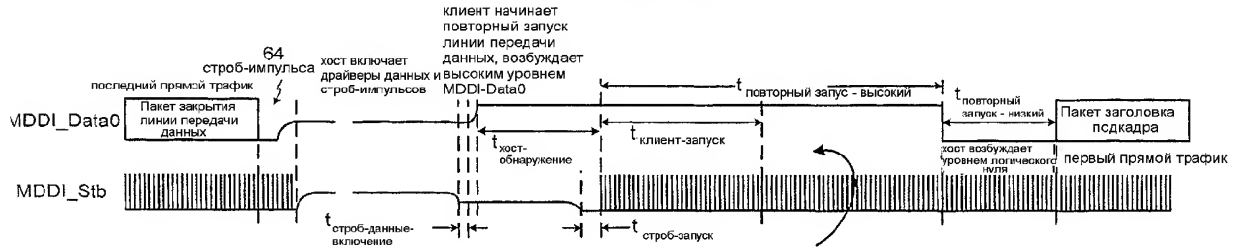
Временные соотношения перехода линии передачи данных в состояние "спячки"

ФИГ.43а



Иницируемая хостом последовательность пробуждения

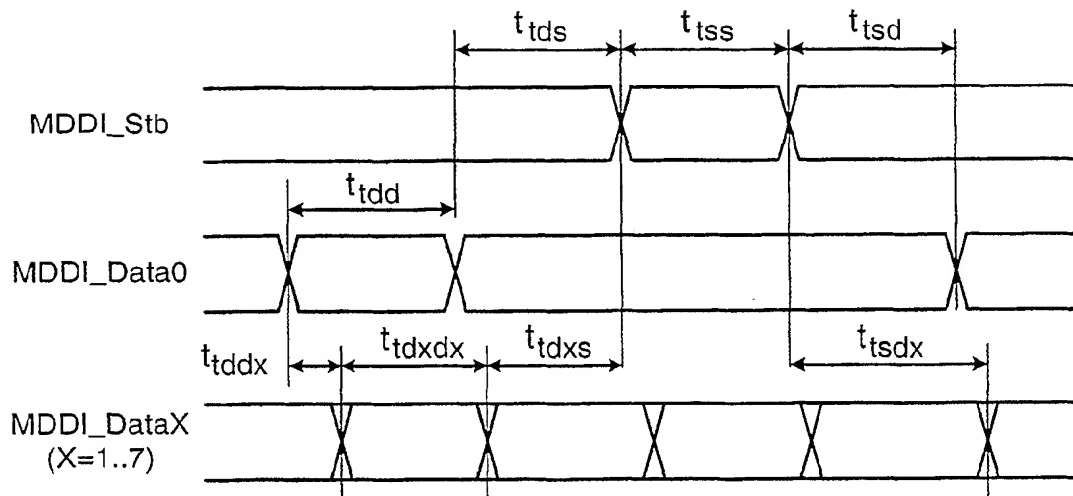
ФИГ.43б



Иницируемая клиентом последовательность пробуждения

Хост отвечает возбуждением уровнем логической единицы, затем клиент прекращает возбуждение MDDI_Data0

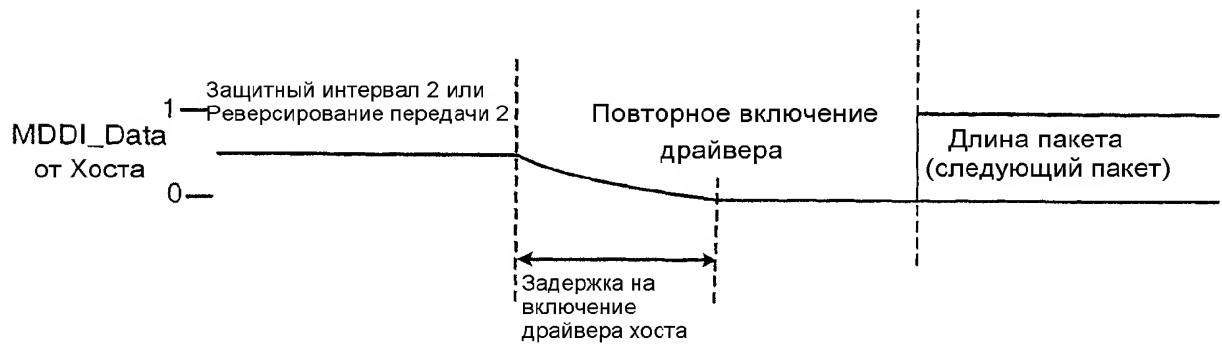
ФИГ.43с



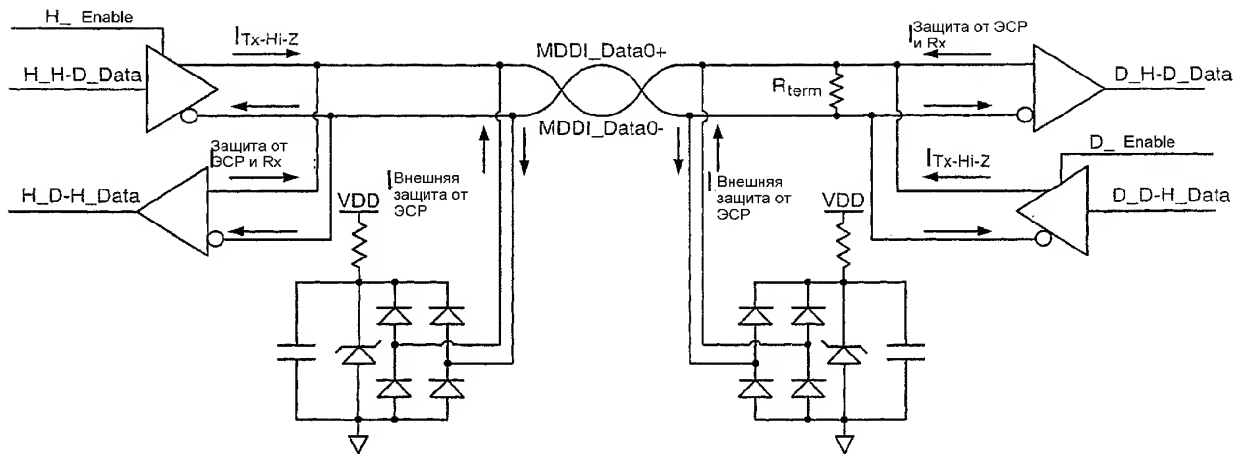
ФИГ.44



ФИГ.45



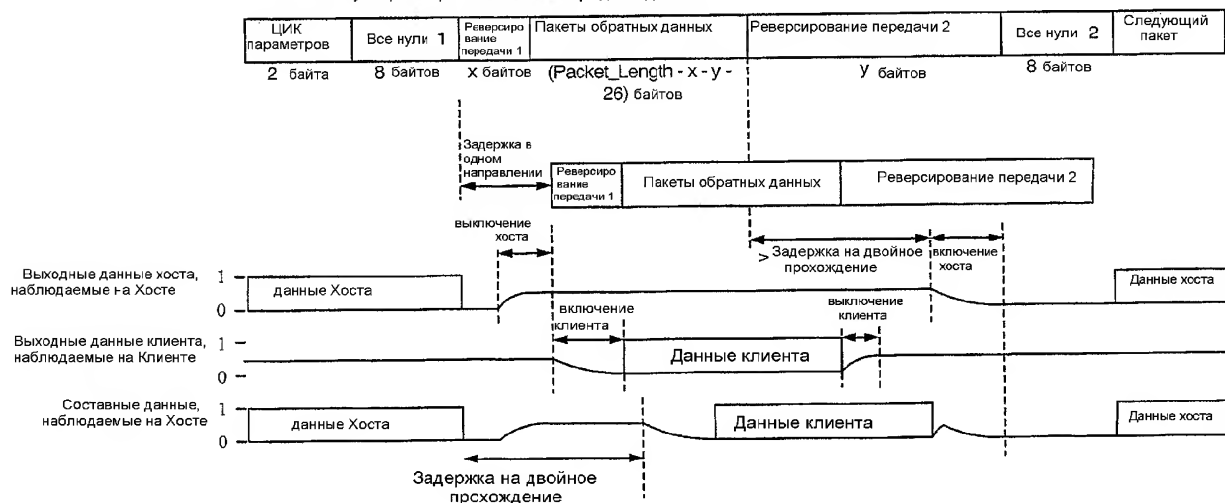
ФИГ.46



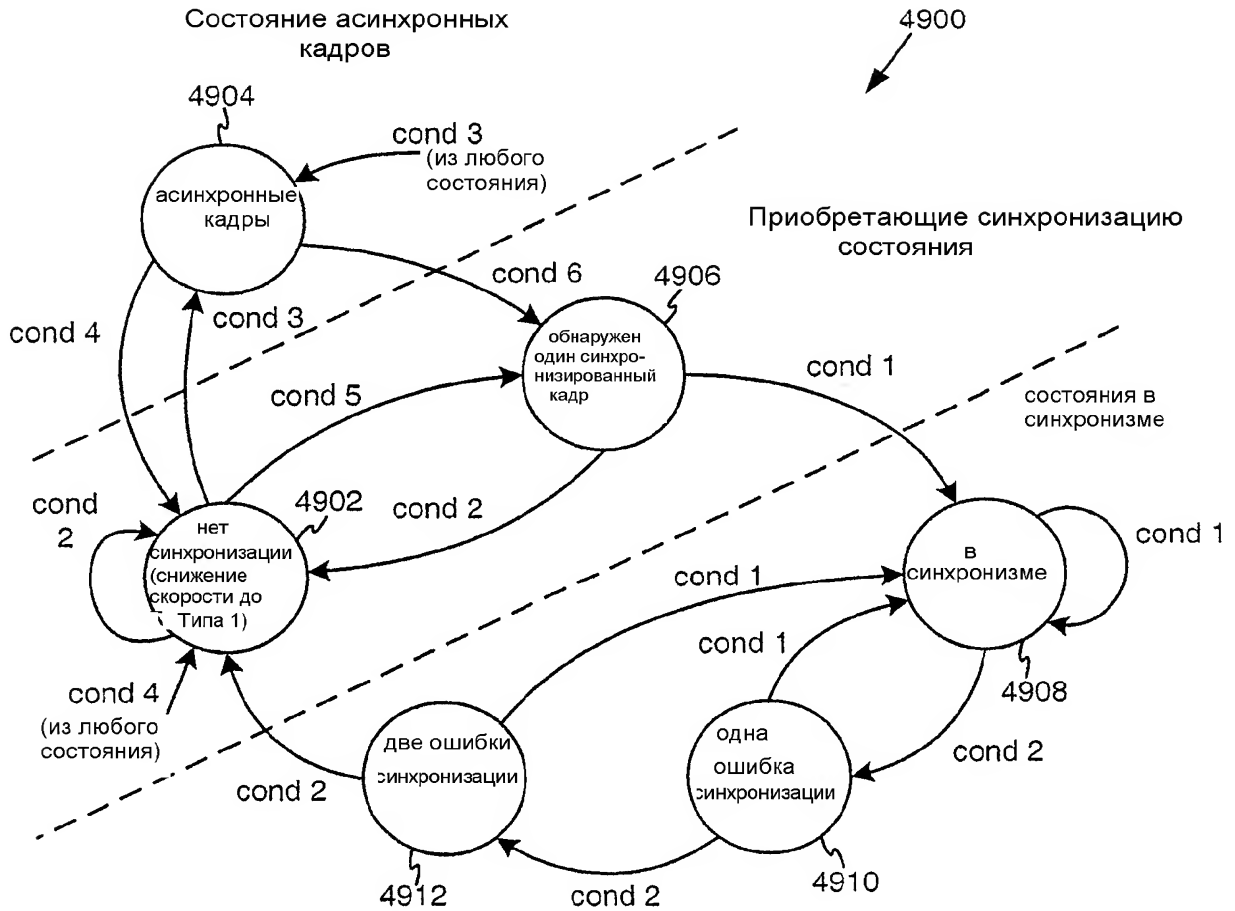
term - согласованной нагрузки

ФИГ.47

Пакет инкапсуляции обратной линии передачи данных

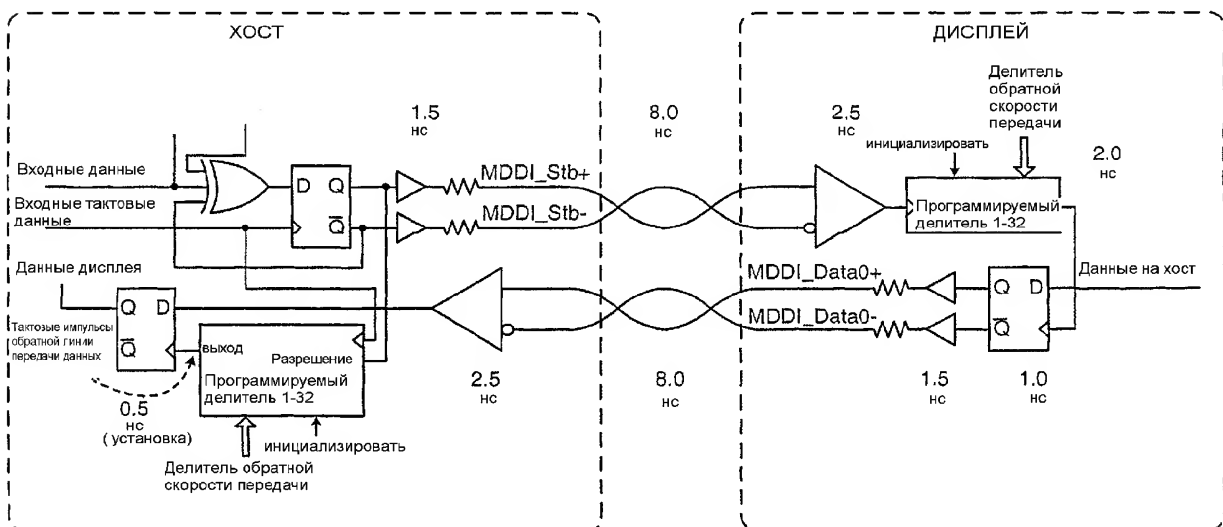


ФИГ.48

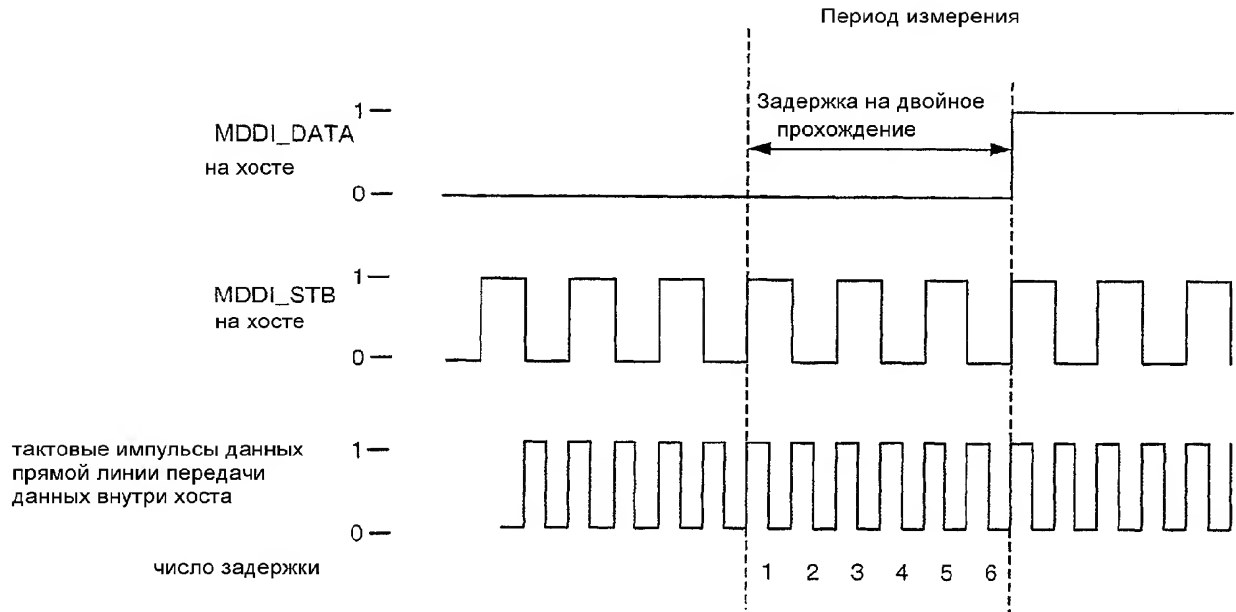


- cond 1 = пакет заголовка подкадра и ЦИК без ошибок на границе кадра, Длина кадра > 0
 cond 2 = нет комбинации синхронизации или ЦИК с ошибками на границе кадра
 cond 3 = обнаружена комбинация синхронизации, Длина кадра = 0
 cond 4 = принят пакет закрытия линии передачи данных
 cond 5 = обнаружена комбинация синхронизации, Длина кадра > 0
 cond 6 = пакет заголовка кадра и ЦИК без ошибок, Длина кадра > 0

ФИГ.49

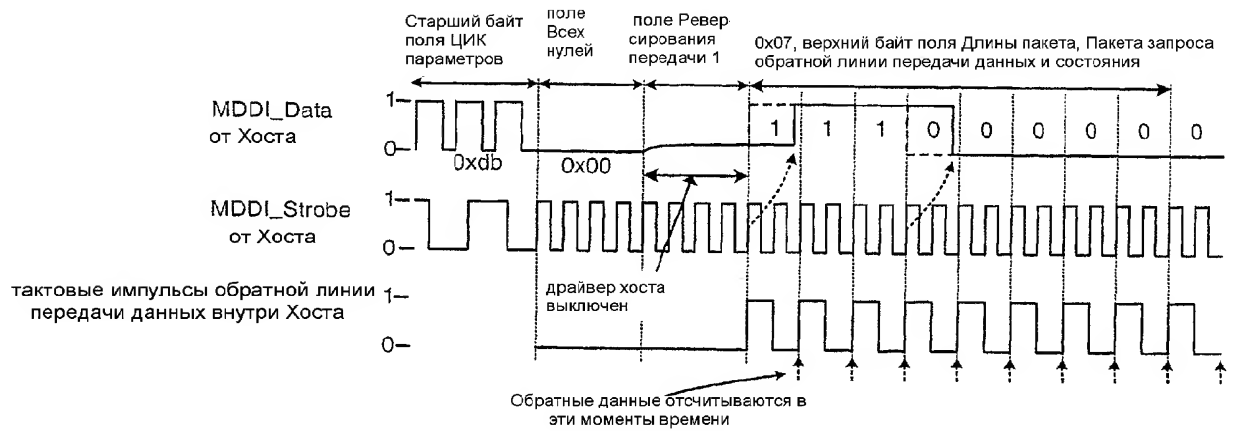


ФИГ.50



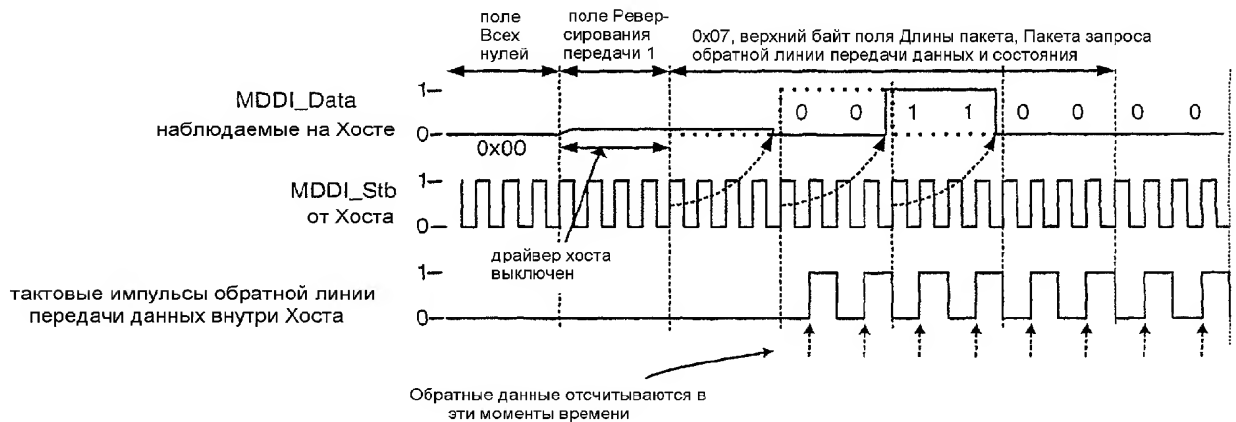
ФИГ.51

Пакеты обратной линии передачи данных



ФИГ.52A

Пакеты обратной линии передачи данных

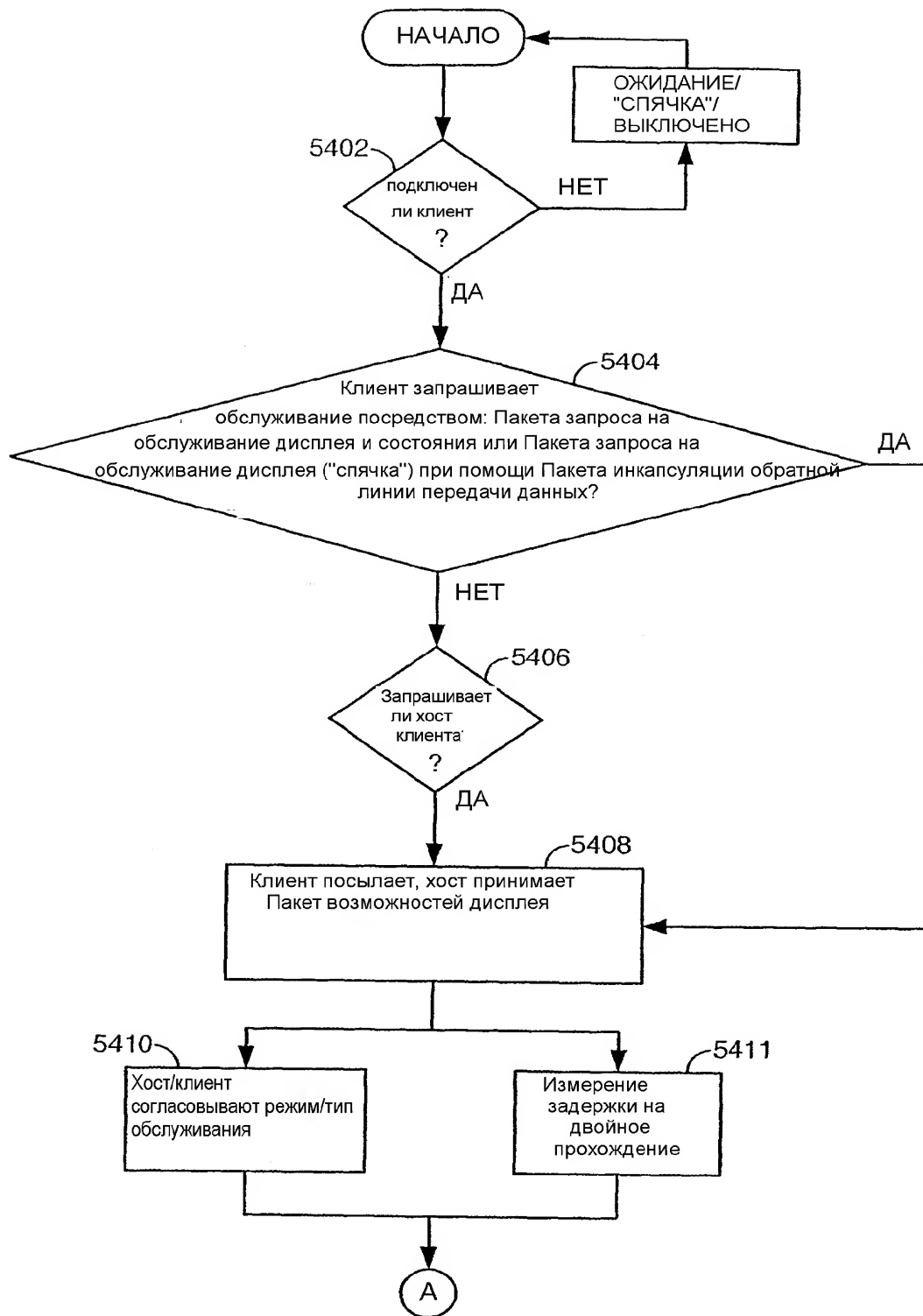


ФИГ.52B

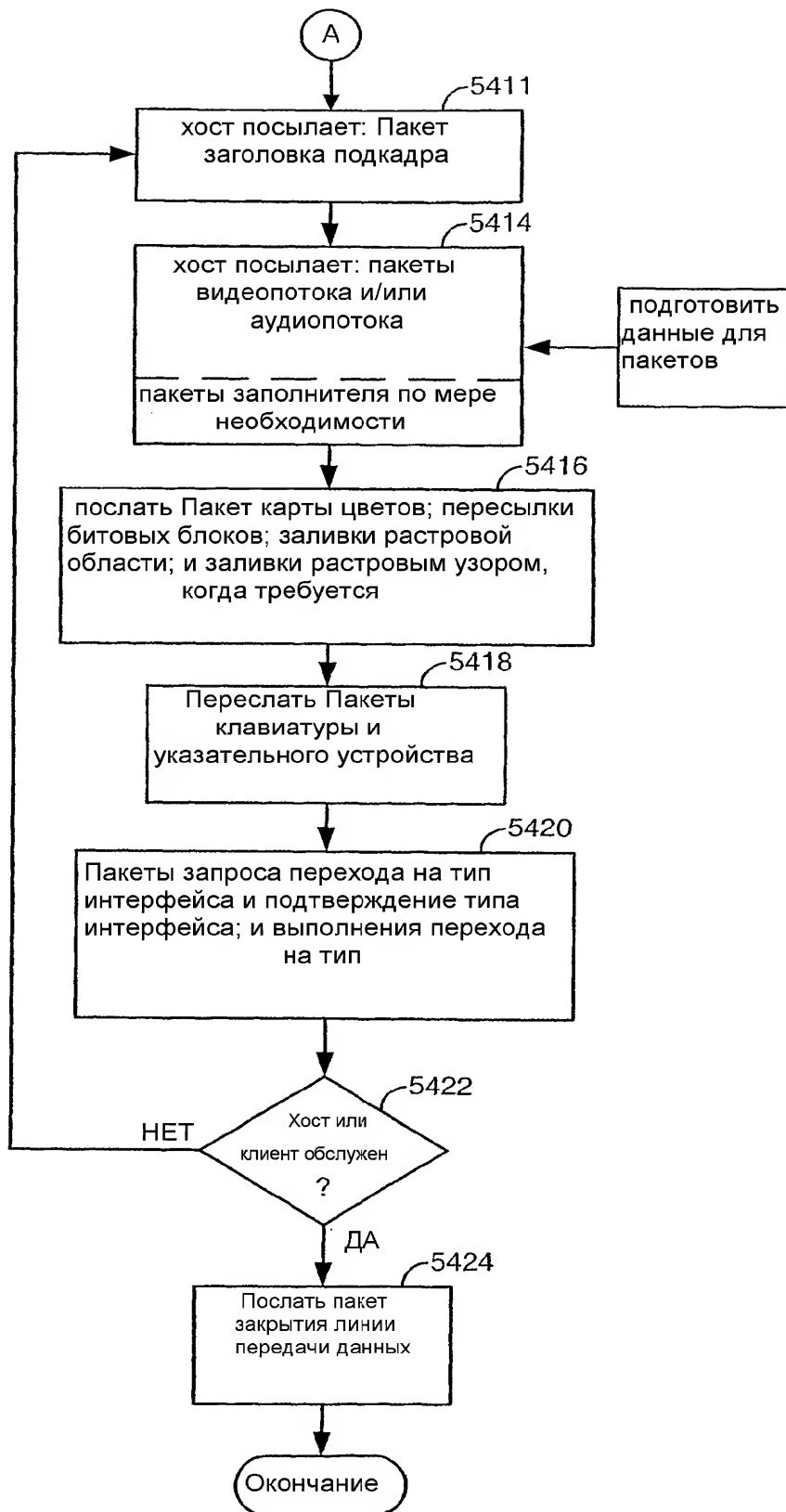
Делитель обратной
скорости передачи



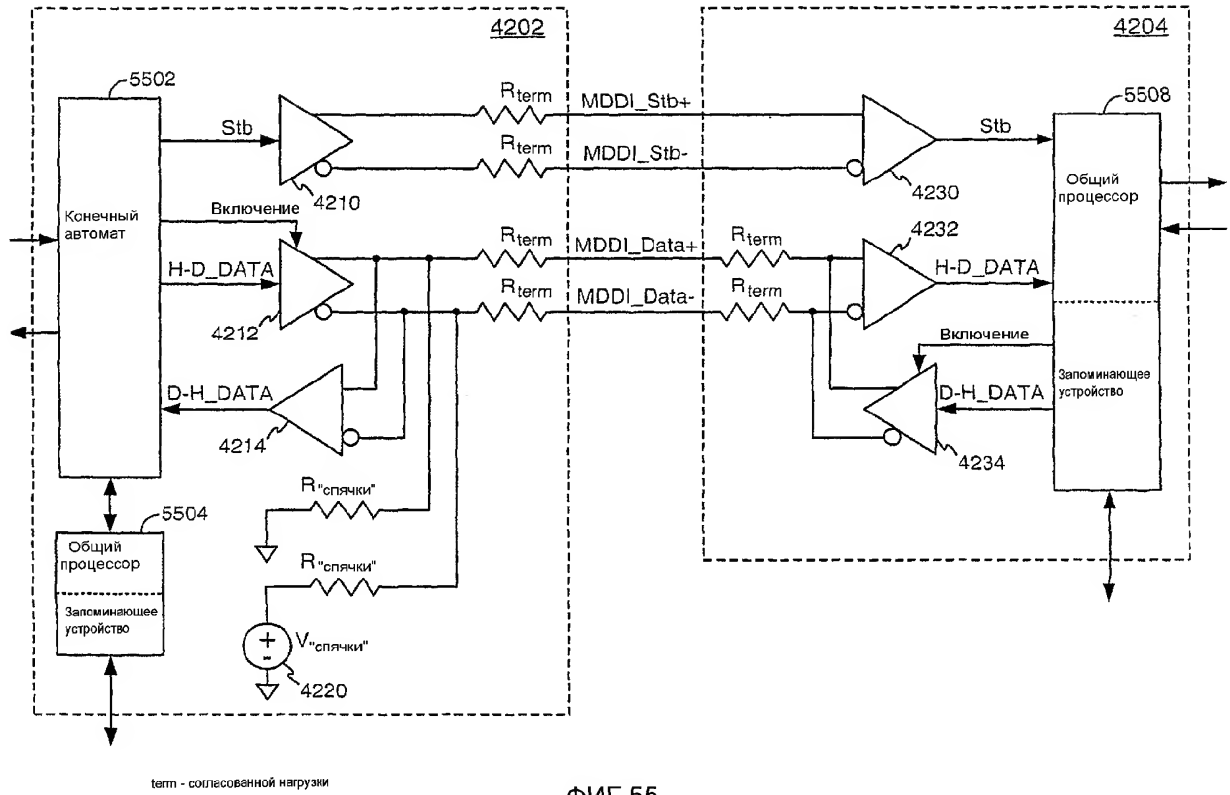
ФИГ.53



ФИГ.54А



ФИГ.54В

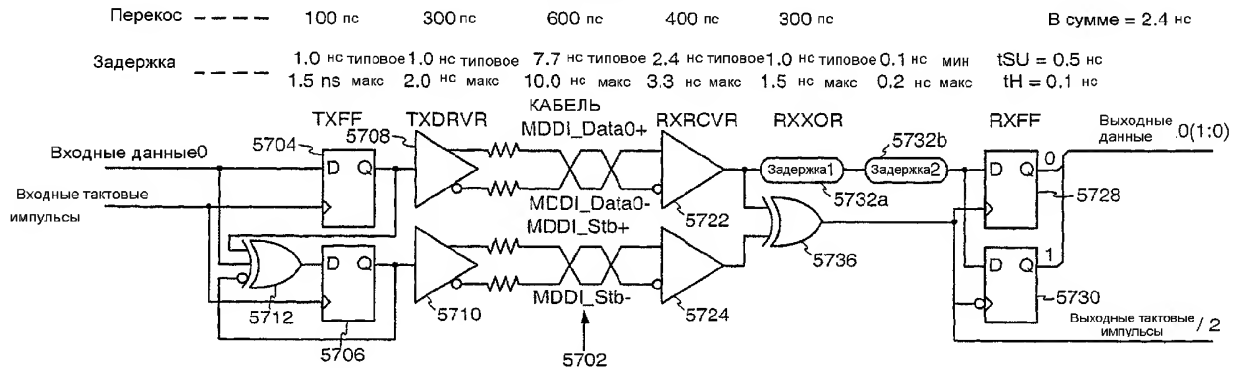


ФИГ.55

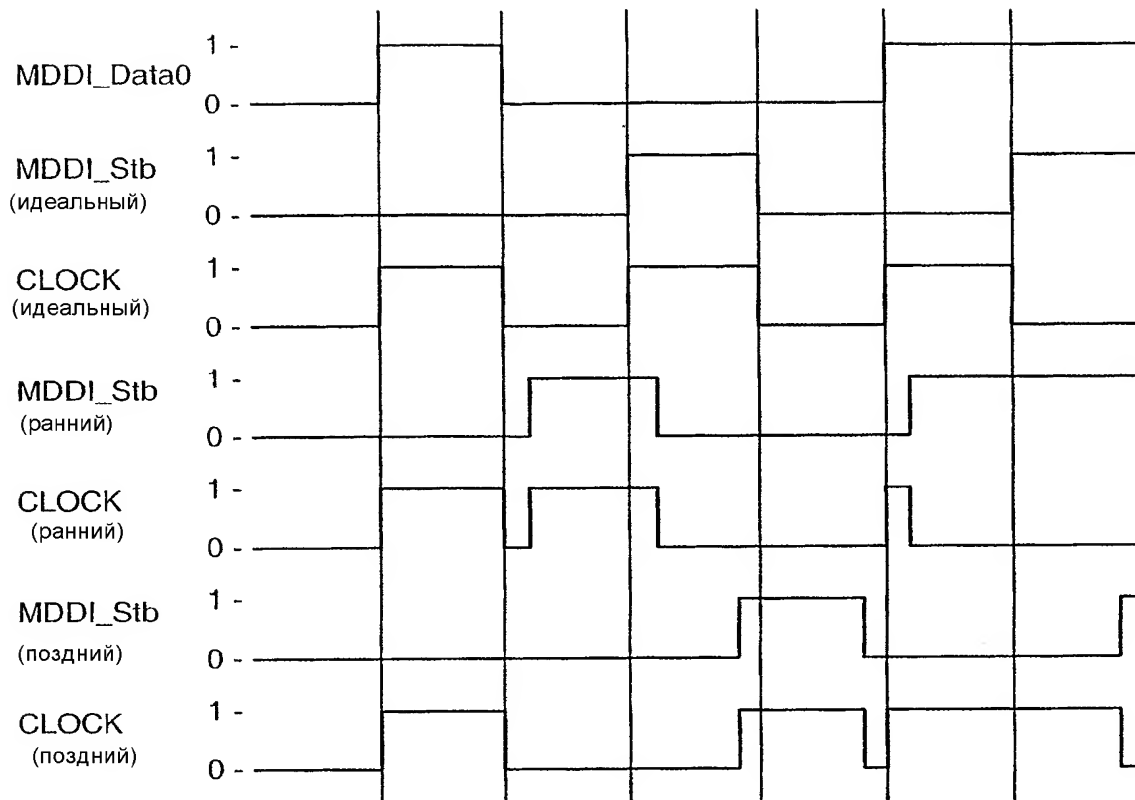
Пакет калибровки перекоса прямой линии передачи данных

Длина пакета	Тип пакета = 83	ИД hClient	ЦИК параметров	Все нули 1	Последовательности данных калибровки	Все нули 2
2 байта	2 байта	2 байта	2 байта	8 байтов	Длина пакета - 22 байта	8 байтов

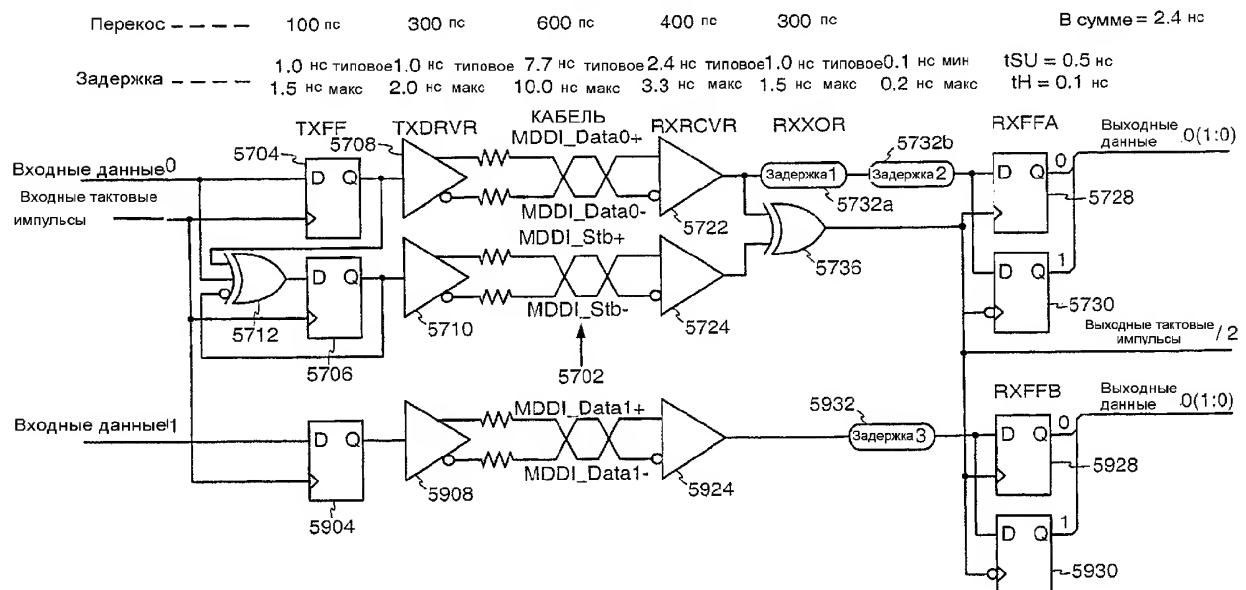
ФИГ.56



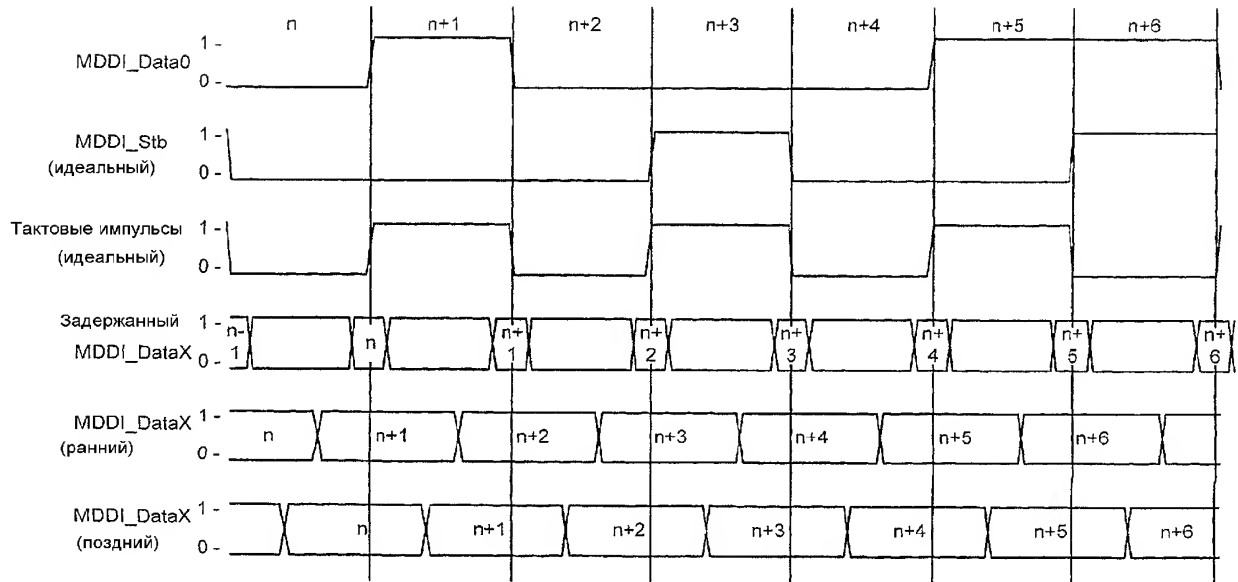
ФИГ.57



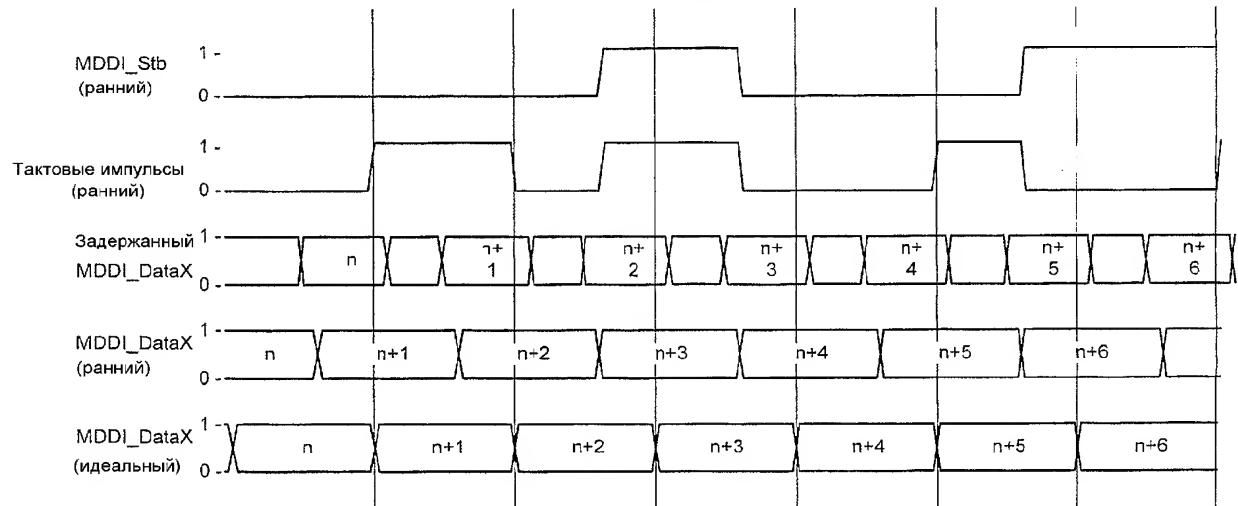
ФИГ.58



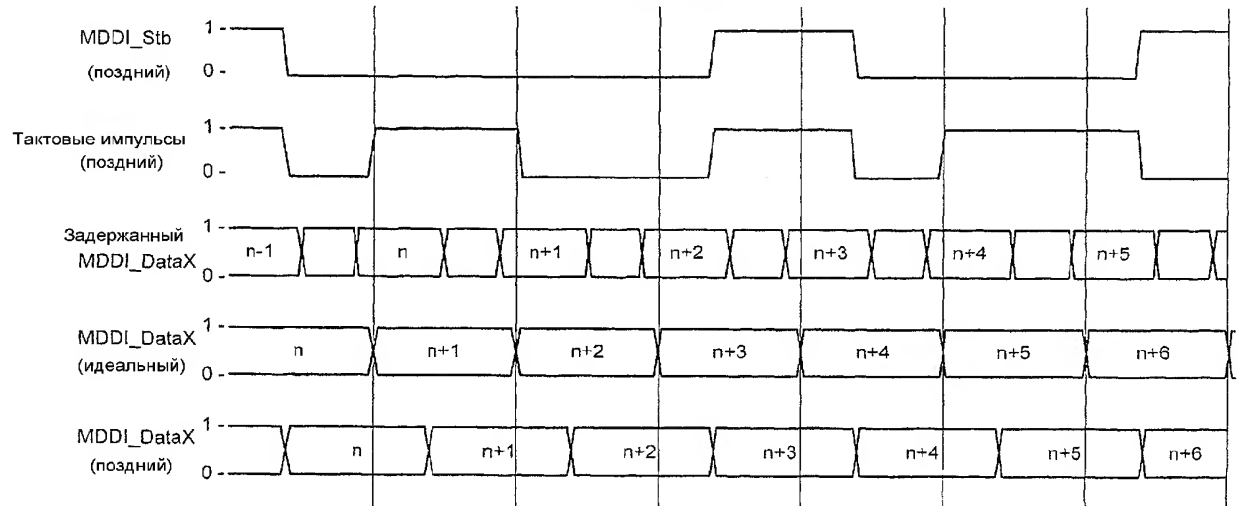
ФИГ.59



ФИГ.60А

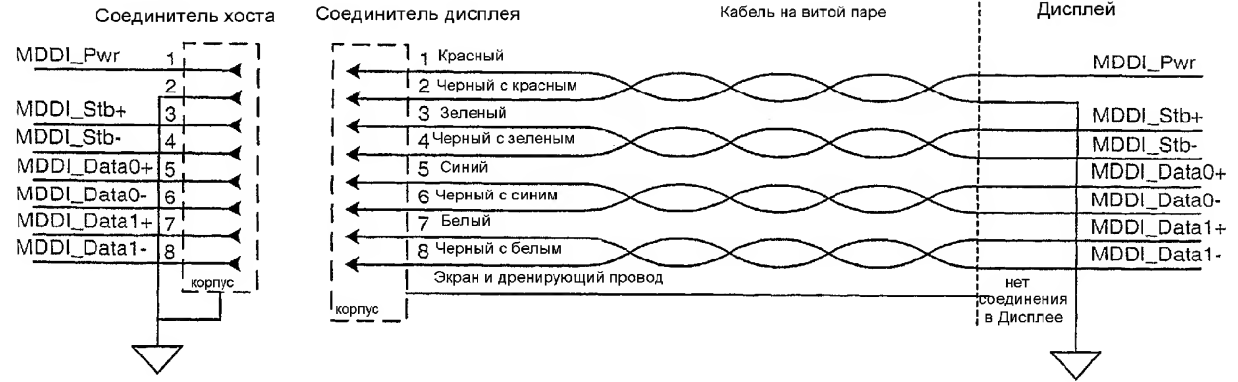


ФИГ.60В



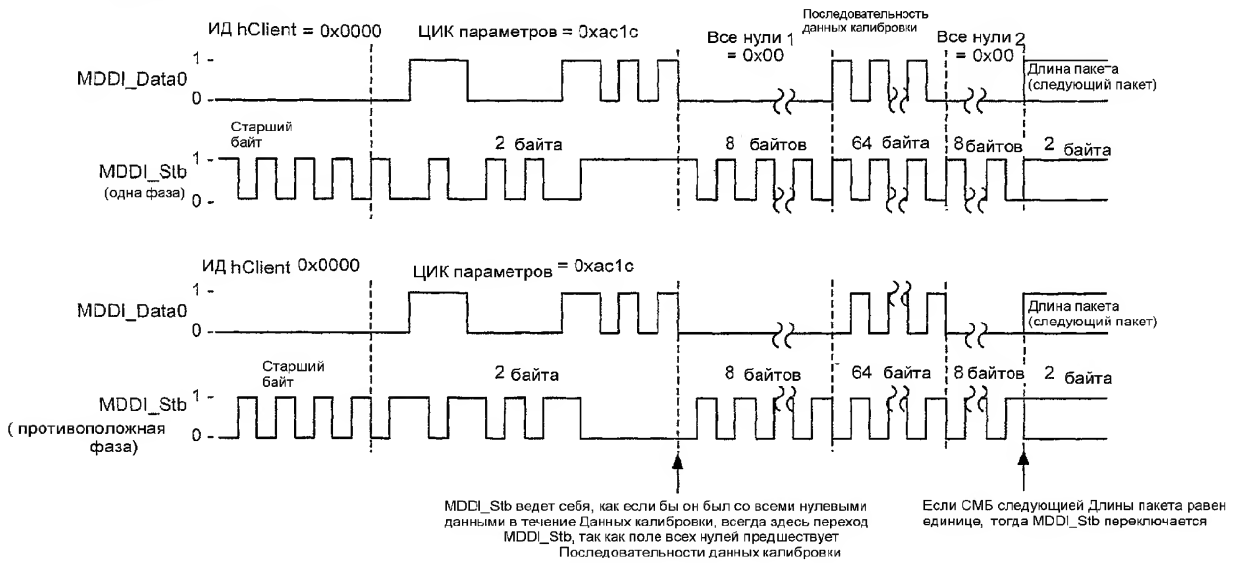
ФИГ.60С

ХОСТ



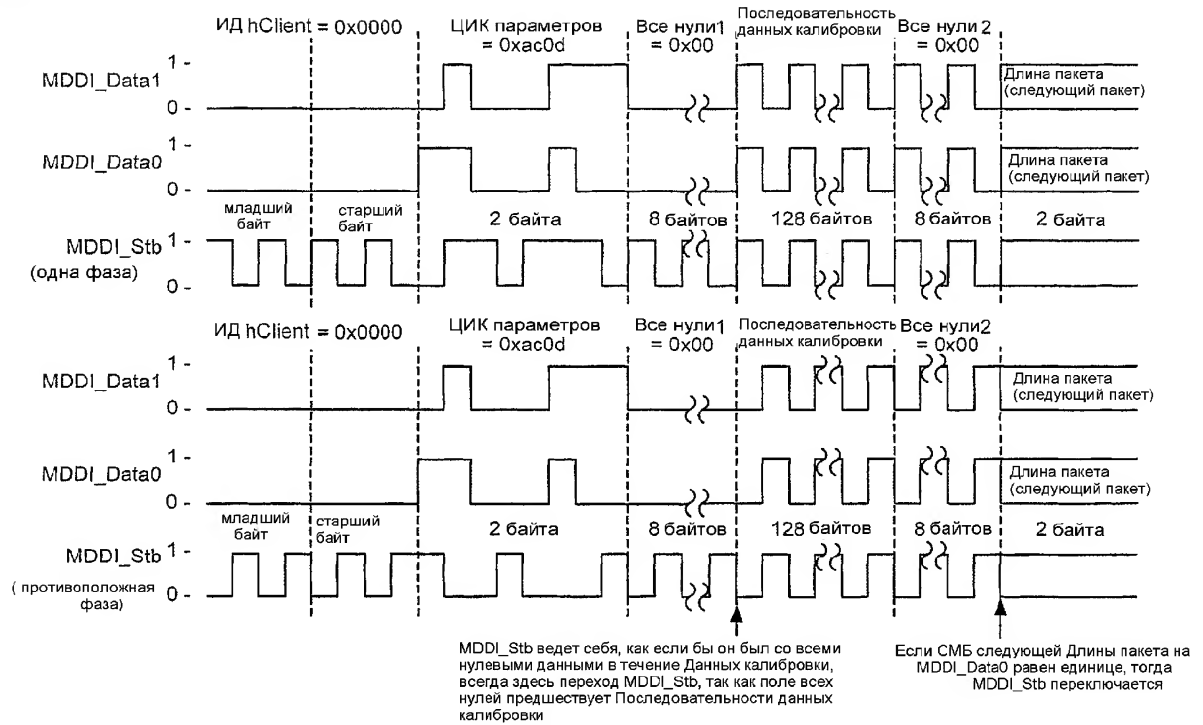
ФИГ.61

Пример данных Типа-I

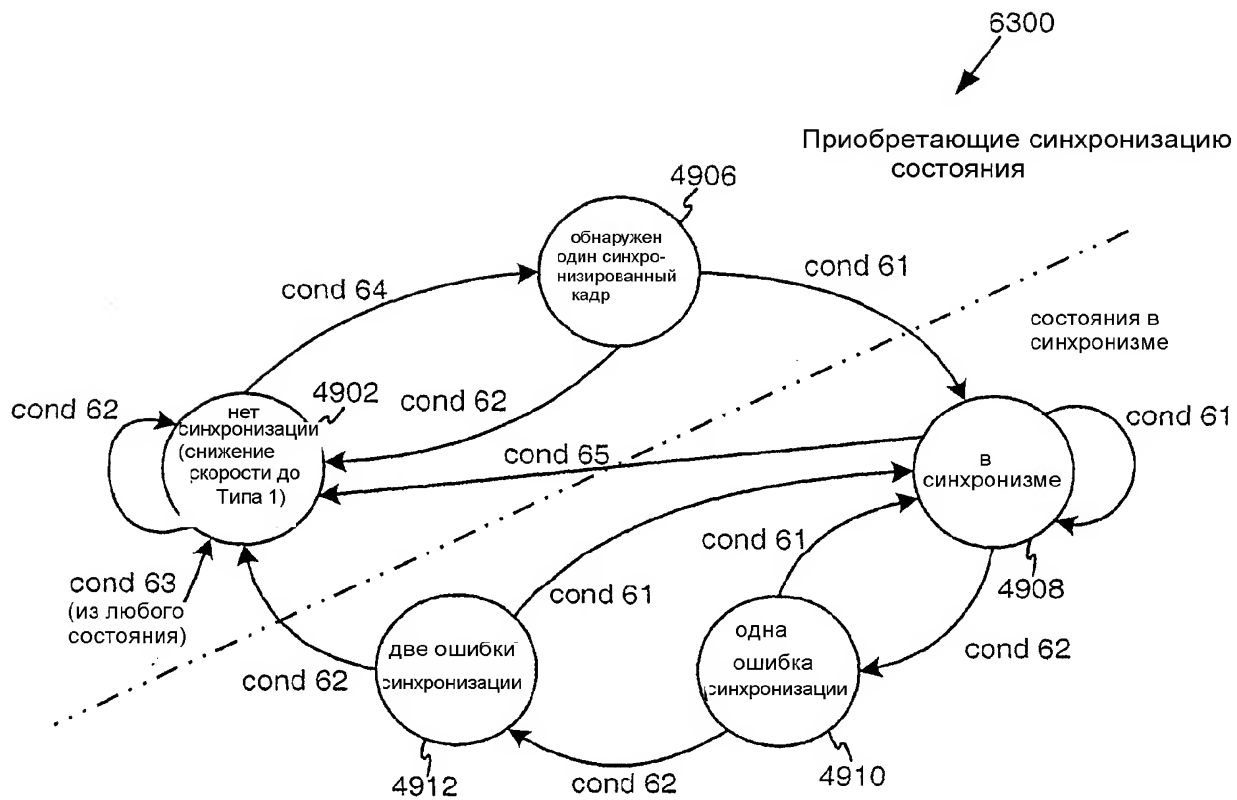


ФИГ.62А

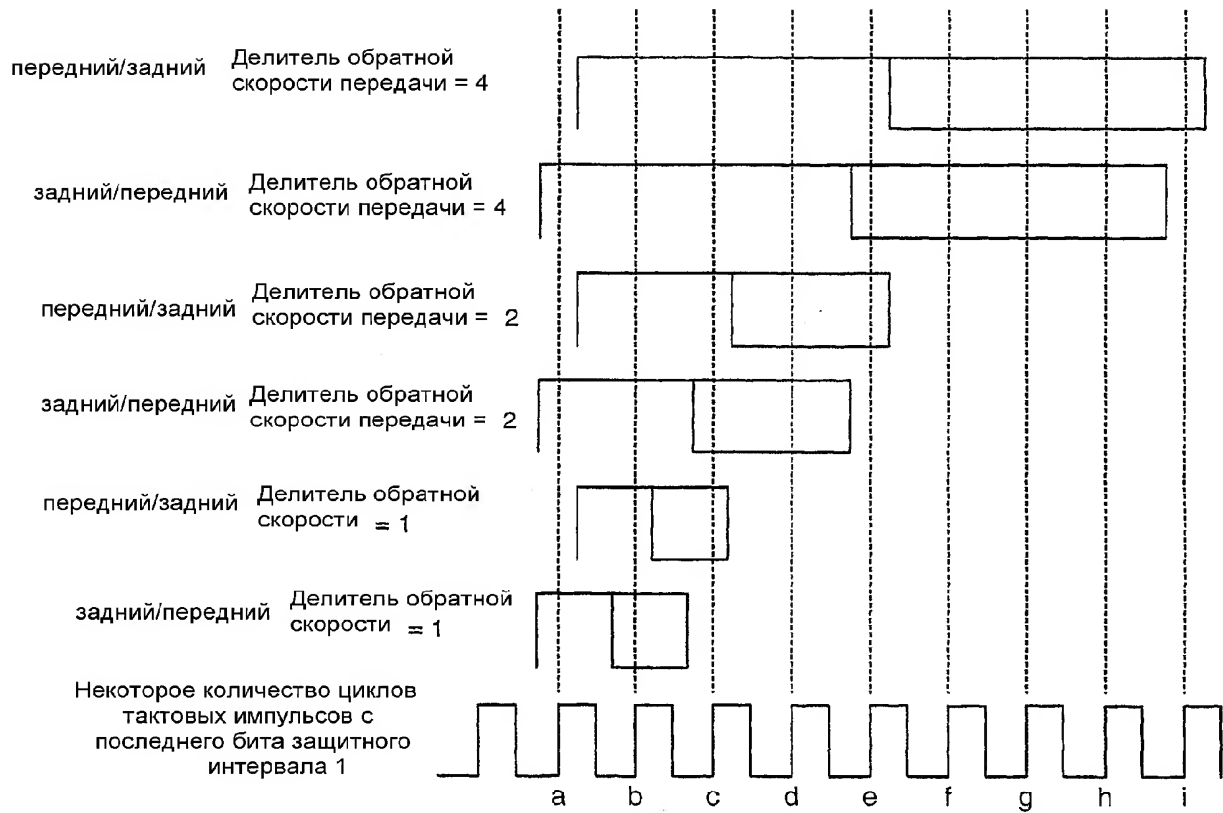
Пример данных Типа-II



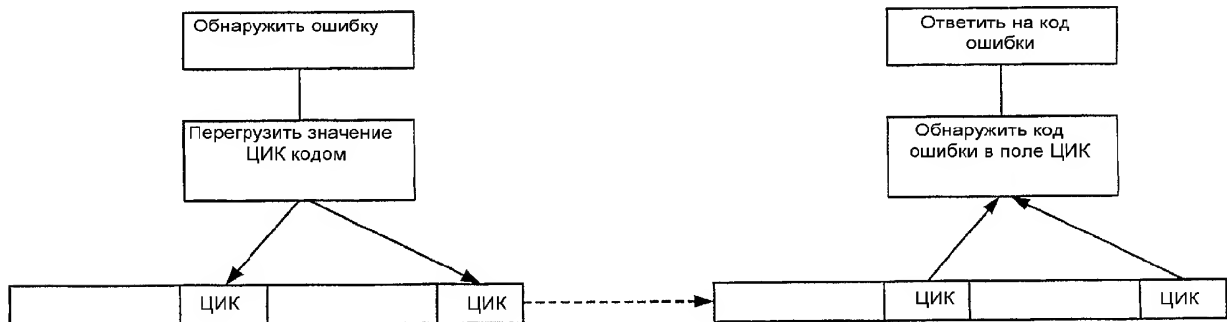
ФИГ.62В



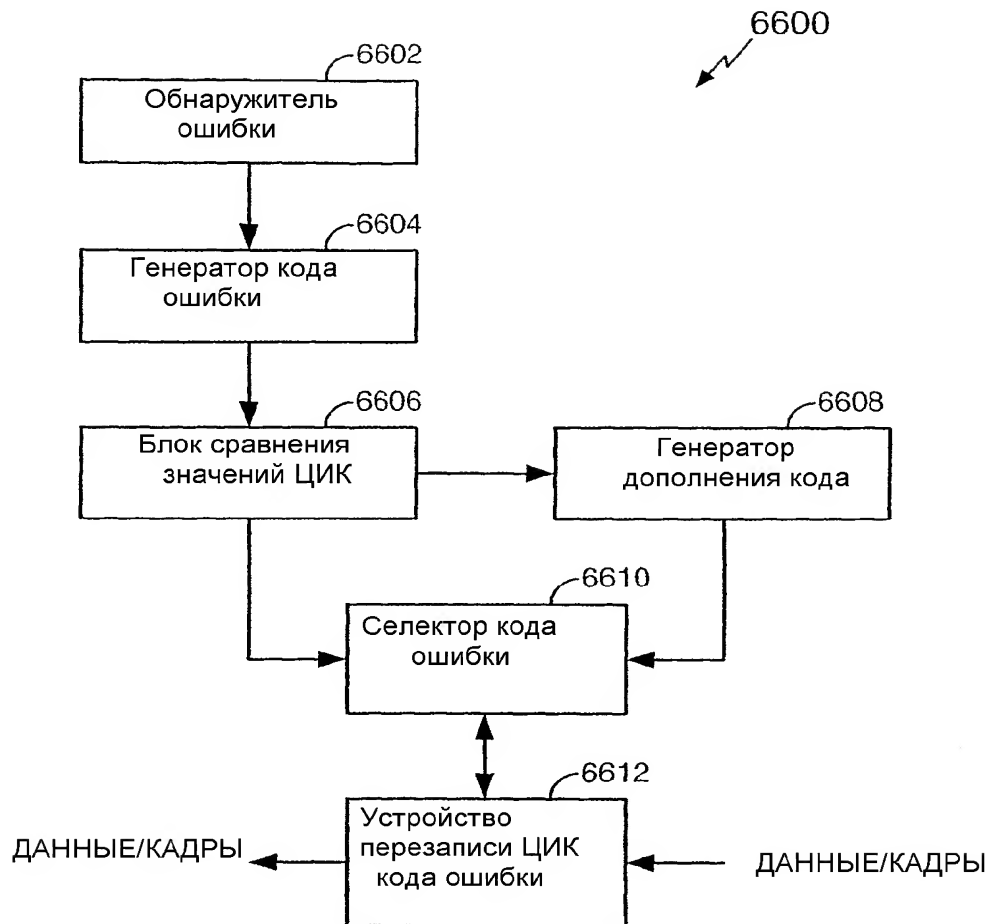
ФИГ.63



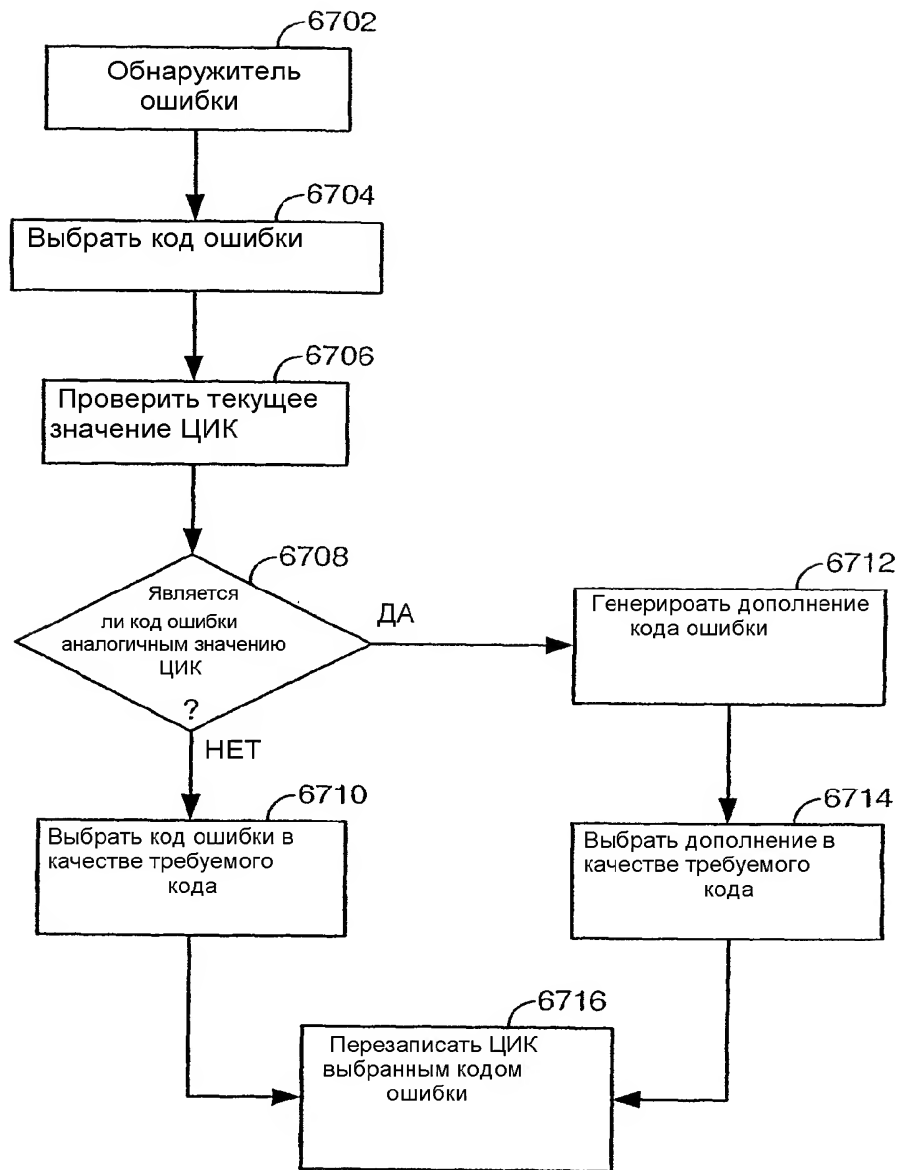
ФИГ.64



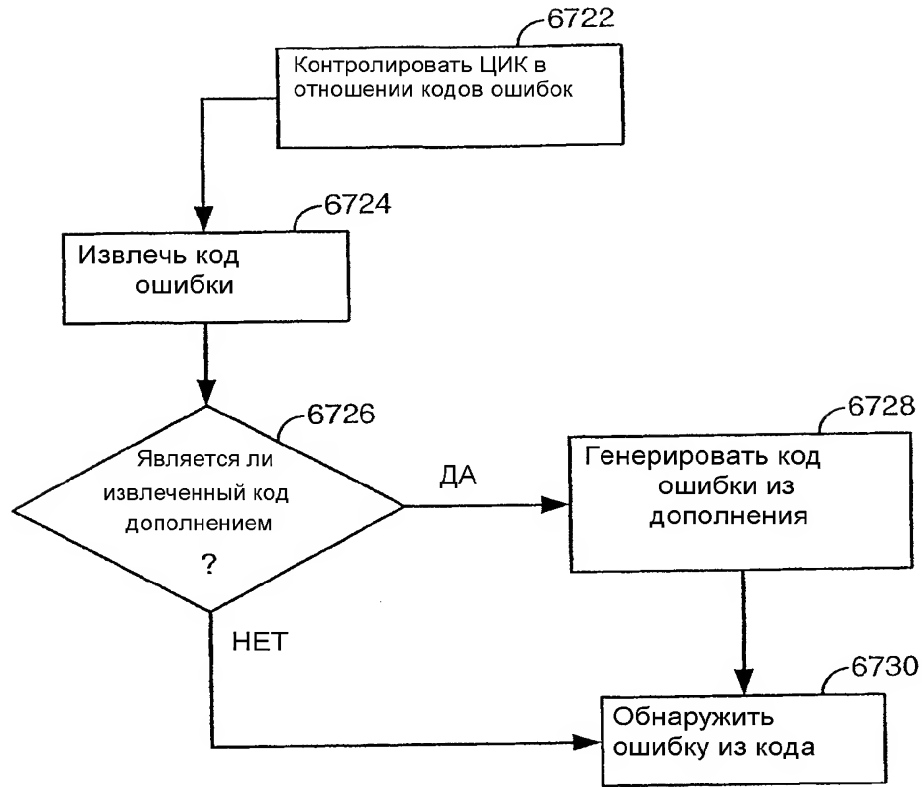
ФИГ.65



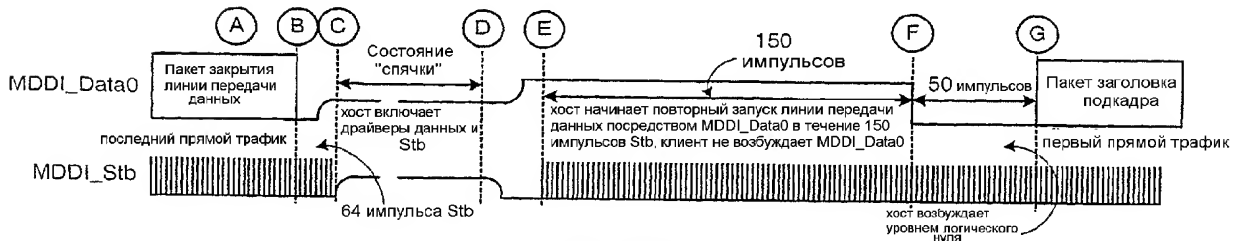
ФИГ.66



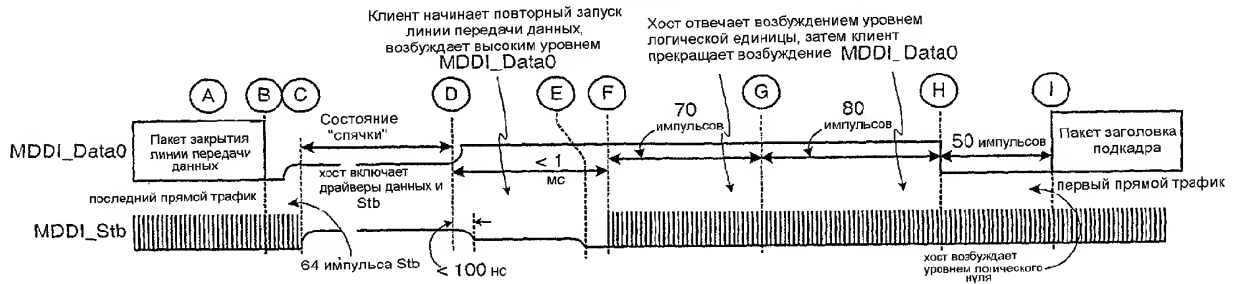
ФИГ.67А



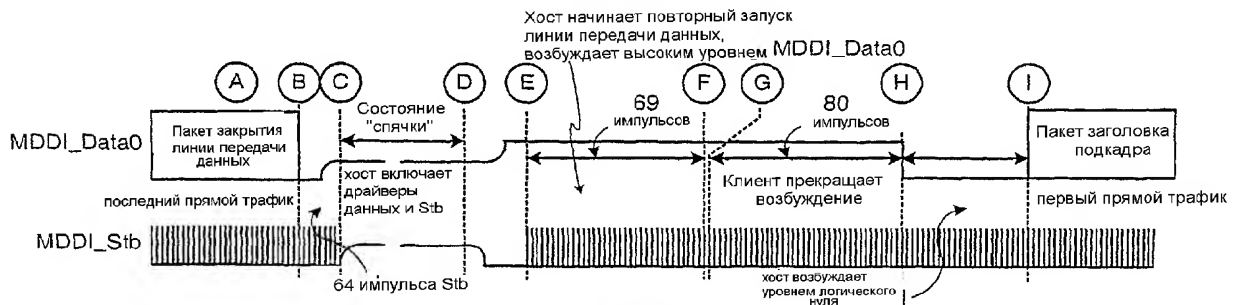
ФИГ.67B



ФИГ.68A



ФИГ.68B



ФИГ.68C

Пакет запроса свойств ВПУ

Длина пакета	Тип пакета = 128	ИД hClient	Код ВПУ НКУМ	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.69

Пакет ответа свойств ВПУ

Длина пакета	Тип пакета = 129	ИД cClient	Версия НКУМ	Порядковый номер ответа	Количество свойств в списке	Список ответа свойств ВПУ	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	(Packet_Length - 12 байтов)	2 байта

ФИГ.70

Элемент списка ответа свойств ВПУ

Код ВПУ НКУМ	Код результата	Максимальное значение	Текущее значение
2 байта	2 байта	4 байта	4 байта

ФИГ.71

Пакет установки свойств ВПУ

Длина пакета	Тип пакета = 130	ИД hClient	Код ВПУ НКУМ	Количество значений в списке	Список значений управления	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта	(Packet_Length - 10 байтов)	2 байта

ФИГ.72

Пакет запроса действительных параметров

Длина пакета	Тип пакета = 131	ИД hClient	Код ВПУ НКУМ	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.73

Пакет ответа действительных параметров

Длина пакета	Тип пакета = 132	ИД cClient	Код ВПУ НКУМ	Код ответа	Порядковый номер ответа	Количество значений в списке	Список ответа параметров ВПУ	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	(Packet_Length - 14 байтов)	2 байта

ФИГ.74

Пакет возможностей масштабируемого видеопотока

Длина пакета	Тип пакета = 143	ИД cClient	Максимальное количество потоков	Максимальный размер источника по оси X	Максимальный размер источника по оси Y
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта

Возможности формата RGB	Возможности монохромного формата	Резервированное 1	Возможности формата Y Cr Cb	Биты возможностей	Резервированное 2	ЦИК
2 байта	1 байт	1 байт	2 байта	1 байт	1 байт	2 байта

ФИГ.75

Пакет установки масштабируемого видеопотока

Длина пакета	Тип пакета = 136	ИД hClient	ИД потока	Дескриптор формата видеоданных	Атрибуты данных пикселей
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта

Левый край по оси X	Верхний край по оси Y	Правый край по оси X	Нижний край по оси Y	Размер изображения по оси X	Размер изображения по оси Y	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.76

Пакет подтверждения масштабируемого видеопотока

Длина пакета	Тип пакета = 137	ИД cClient	ИД потока	Код подтверждения	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.77

Пакет масштабируемого видеопотока

Длина пакета	Тип пакета = 18	ИД hClient	ИД потока	Атрибуты данных пикселей	Число пикселей	ЦИК параметров	Данные пикселей	ЦИК данных пикселей
2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	2 байта	Длина пакета - 12 байтов	2 байта

ФИГ.78

Пакет запроса заданного состояния

Длина пакета	Тип пакета = 138	ИД hClient	ИД пакета состояния	ЦИК
2 байта	2 байта	2 байта	2 байта	2 байта

ФИГ.79

Пакет списка ответа действительных состояний

Длина пакета	Тип пакета = 139	ИД cClient	Количество значений в списке	Список ответа действительных параметров	ЦИК
2 байта	2 байта	2 байта	2 байта	Длина пакета - 8 байтов	2 байта

ФИГ.80

Пакет параметров задержки на обработку пакета

Длина пакета	Тип пакета = 140	ИД cClient	Количество элементов списка	Список параметров задержки	ЦИК
2 байта	2 байта	2 байта	2 байта	Длина пакета - 8 байтов	2 байта

ФИГ.81А

Элемент списка параметров задержки

Тип пакета для задержки	Задержка пиксела	Задержка пиксела по горизонтали	Задержка пиксела по вертикали	Фиксированная задержка
2 байта	1 байт	1 байт	1 байт	1 байт

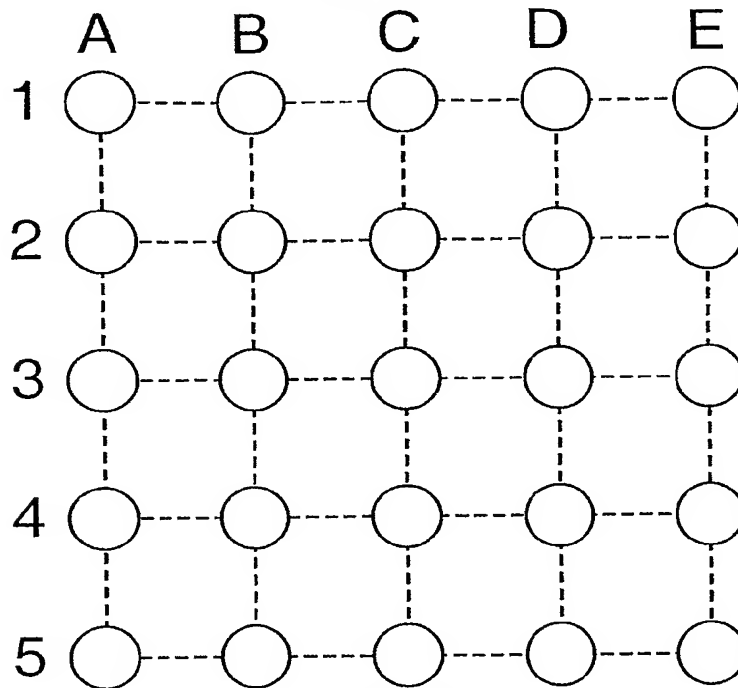
ФИГ.81В

Пакет возможностей персонального дисплея

Длина пакета	Тип пакета = 141	ИД cClient	Схема расположения субпикселей	Форма пикселей	Поле зрения по горизонтали	Поле зрения по вертикали	Пересечение зрительных осей
2 байта	2 байта	2 байта	1 байт	1 байт	1 байт	1 байт	1 байт

Перекрытие левого/правого изображения	Сквозной просмотр	Максимальная яркость	Оптические возможности	Минимальное РМЗ	Максимальное РМЗ	Список точек кривизны поля (25 2-байтовых значений)	ЦИК
1 байт	1 байт	1 байт	2 байта	1 байт	1 байт	50 байтов	2 байта

ФИГ.82



ФИГ.83

Пакет отчета об ошибках клиента

Длина пакета	Тип пакета = 142	ИД cClient	Количество элементов списка	Список кодов ошибок	ЦИК
2 байта	2 байта	2 байта	2 байта	Длина пакета - 8 байтов	2 байта

ФИГ.84А

Элемент списка отчета об ошибках

Код ошибки дисплея	Подкод ошибки
2 байта	2 байта

ФИГ.84В

Пакет идентификации клиента

Длина пакета	Тип пакета = 144	ИД cClient	Неделя изготовления	Год изготовления	Длина названия изготовителя	Длина названия продукта	Длина серийного номера
2 байта	2 байта	2 байта	1 байт	1 байт	2 байта	2 байта	2 байта

Строка названия изготовителя	Строка названия продукта	Строка серийного номера	ЦИК
Длина названия изготовителя в байтах	Длина названия изготовителя в байтах	Длина названия изготовителя в байтах	2 байта

ФИГ.85

Пакет возможностей альтернативного дисплея

Длина пакета	Тип пакета = 145	ИД cClient	Количество альтернативных дисплеев	Резервированное 1	Ширина растрового изображения	Высота растрового изображения	Ширина окна дисплея	Высота окна дисплея
2 байта	2 байта	2 байта	1 байт	1 байт	2 байта	2 байта	2 байта	2 байта
Разрядность карты цветов формата RGB	Возможность формата RGB	Возможности монохромного формата	Резервированное 2	Возможности формата Y Cb Cr	Возможности свойств дисплея	Резервированное 3	ЦИК	
2 байта	2 байта	1 байт	1 байт	2 байта	1 байт	1 байт	2 байта	

ФИГ.86

Пакет доступа к регистрам

Длина пакета	Тип пакета = 146	ИД bClient	Флаги считыва- ния/записи	Адрес регистра	ЦИК параметров	Список данных регистров	ЦИК
2 байта	2 байта	2 байта	2 байта	4 байта	2 байта	Длина пакета - 14 байтов	2 байта

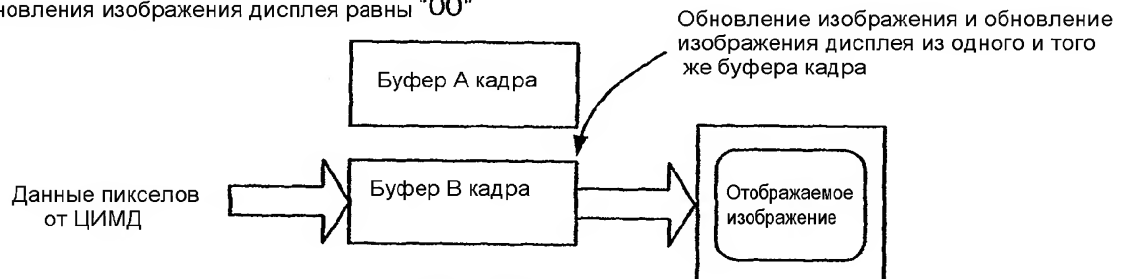
ФИГ.87

Биты обновления изображения дисплея равны "01"



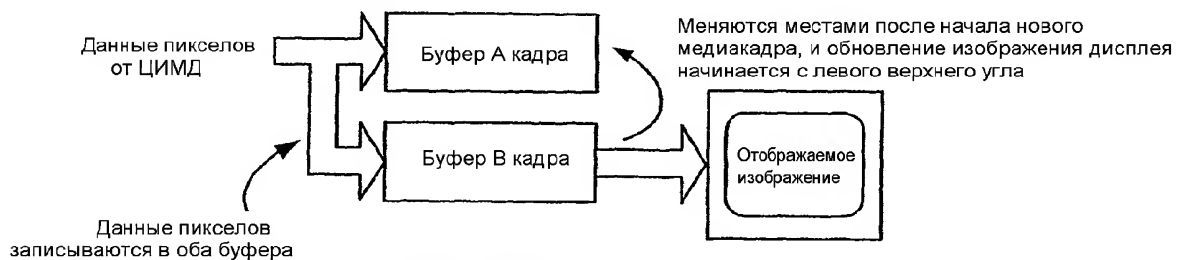
ФИГ.88А

Биты обновления изображения дисплея равны "00"



ФИГ.88В

Биты обновления изображения дисплея равны "11"



ФИГ.88С

Два буфера, обновление изображения дисплея быстрее,
чем пересылка изображения

пересылка изображения

кадр 1, бусфер А 20% 40% 60% 80%	кадр 2, бусфер В 20% 40% 60% 80%	кадр 3, бусфер А 20% 40% 60% 80%	кадр 4, бусфер В 20% 40% 60% 80%	кадр 5, бусфер А 20% 40% 60% 80%	кадр 6, бусфер В 20% 40% 60% 80%
-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------

Обновление изображения дисплея

[illegible]

Два буфера, обновление изображения дисплея значительно быстрее,
чем пересылка изображения

пересылка изображения

кадр 1, буфер А 20% 40% 60% 80%	кадр 2, буфер В 20% 40% 60% 80%	кадр 3, буфер А 20% 40% 60% 80%	кадр 4, буфер В 20% 40% 60% 80%	кадр 5, буфер А 20% 40% 60% 80%	кадр 6, буфер В 20% 40% 60% 80%
------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------

Обновление изображения дисплея

[illegible]

ФИГ.89

Два буфера, обновление изображения дисплея медленнее,
чем пересылка изображения

пересылка изображения

кадр 1, буфер А 20% 40% 60% 80%	кадр 2, буфер В 20% 40% 60% 80%	кадр 3, буфер А 20% 40% 60% 80%	кадр 4, буфер В 20% 40% 60% 80%	кадр 5, буфер А 20% 40% 60% 80%	кадр 6, буфер В 20% 40% 60% 80%
------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------

Обновление изображения дисплея

кадр 0, буфер В				кадры 1 и 3, буфер А				кадр 3, буфер А				кадры 4 и 6, буфер В			
20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%

ФИГ.90

Два буфера, обновление изображения дисплея значительно быстрее, чем пересылка изображения, малое видеоокно

пересылка изображения

кадр 1, буфер А 20% 40% 60% 80%	кадр 2, буфер В 20% 40% 60% 80%	кадр 3, буфер А 20% 40% 60% 80%	кадр 4, буфер В 20% 40% 60% 80%	кадр 5, буфер А 20% 40% 60% 80%	кадр 6, буфер В 20% 40% 60% 80%
------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------

Обновление изображения дисплея

[illegible]

b - буфер

ФИГ.91

Три буфера, обновление изображения дисплея значительно быстрее,
чем пересылка изображения, видеокно любого размера

Пересылка изображения

кадр 1, буфер А 20% 40% 60% 80%	кадр 2, буфер В 20% 40% 60% 80%	кадр 3, буфер С 20% 40% 60% 80%	кадр 4, буфер А 20% 40% 60% 80%	кадр 5, буфер В 20% 40% 60% 80%	кадр 6, буфер С 20% 40% 60% 80%
------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------

Обновление изображения дисплея

[illegible]

b - буфер

ФИГ.92

Три буфера, обновление изображения дисплея медленнее,
чем пересылка изображения

Пересылка изображения

кадр 1, буфер А 20% 40% 60% 80%	кадр 2, буфер В 20% 40% 60% 80%	кадр 3, буфер С 20% 40% 60% 80%	кадр 4, буфер А 20% 40% 60% 80%	кадр 5, буфер В 20% 40% 60% 80%	кадр 6, буфер С 20% 40% 60% 80%	кадр 7 20%
------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	------------------------------------	---------------

Обновление изображения дисплея

кадр 0, буфер С				кадры 1 и 4, буфер А				кадр 3, буфер С				кадры 4 и 7, буфер А			
20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%

ФИГ.93

Один буфер, обновление изображения дисплея быстрее,
чем пересылка изображения

Пересылка изображения

кадр 1 20% 40% 60% 80%	кадр 2 20% 40% 60% 80%	кадр 3 20% 40% 60% 80%	кадр 4 20% 40% 60% 80%	кадр 5 20% 40% 60% 80%	кадр 6 20% 40% 60% 80%
---------------------------	---------------------------	---------------------------	---------------------------	---------------------------	---------------------------

Обновление изображения дисплея

кадр 0 20% 40% 60% 80%	кадр 1 20% 40% 60% 80%	кадр 2 20% 40% 60% 80%	кадр 3 и 2 20% 40% 60% 80%	кадр 3 20% 40% 60% 80%	кадр 4 20% 40% 60% 80%	кадр 5 и 4 20% 40% 60% 80%	кадр 5 20% 40% 60% 80%	кадр 6 20% 40% 60% 80%
---------------------------	---------------------------	---------------------------	-------------------------------	---------------------------	---------------------------	-------------------------------	---------------------------	---------------------------

Один буфер, обновление изображения дисплея значительно быстрее,
чем пересылка изображения

Пересылка изображения

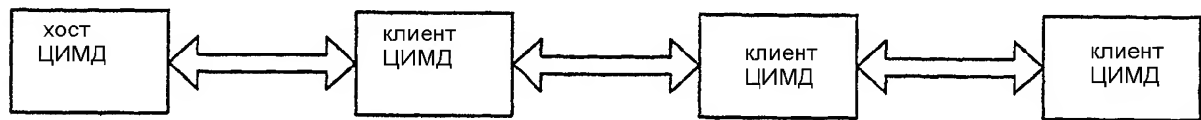
кадр 1 20% 40% 60% 80%	кадр 2 20% 40% 60% 80%	кадр 3 20% 40% 60% 80%	кадр 4 20% 40% 60% 80%	кадр 5 20% 40% 60% 80%	кадр 6 20% 40% 60% 80%
---------------------------	---------------------------	---------------------------	---------------------------	---------------------------	---------------------------

Обновление изображения дисплея

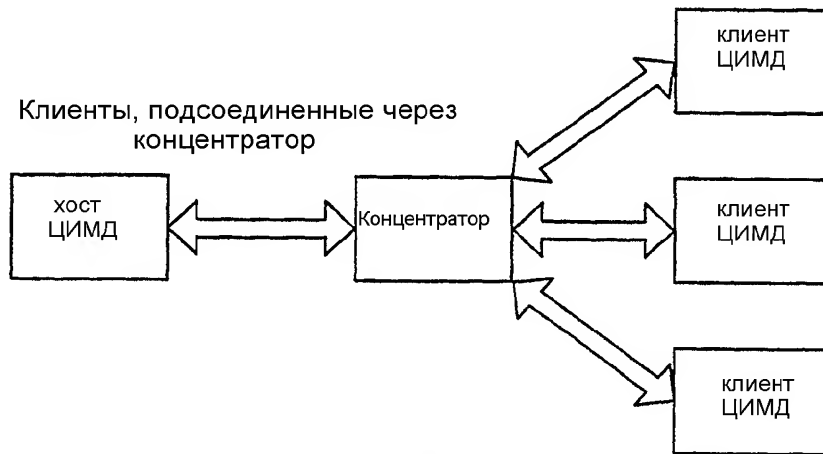
кадр 0 20406080 % % % %	кадр 1 и 0 20406080 % % % %	кадр 1 20406080 % % % %	кадр 2 и 1 20406080 % % % %	кадр 2 20406080 % % % %	кадр 3 и 2 20406080 % % % %	кадр 3 20406080 % % % %	кадр 4 и 3 20406080 % % % %	кадр 4 20406080 % % % %	кадр 5 и 4 20406080 % % % %	кадр 5 и 4 20406080 % % % %	кадр 5 20406080 % % % %	кадр 6 и 5 20406080 % % % %	кадр 6 20406080 % % % %	кадр 6 20406080 % % % %
-------------------------------	-----------------------------------	-------------------------------	-----------------------------------	-------------------------------	-----------------------------------	-------------------------------	-----------------------------------	-------------------------------	-----------------------------------	-----------------------------------	-------------------------------	-----------------------------------	-------------------------------	-------------------------------

ФИГ.94

Конфигурация с последовательным подключением



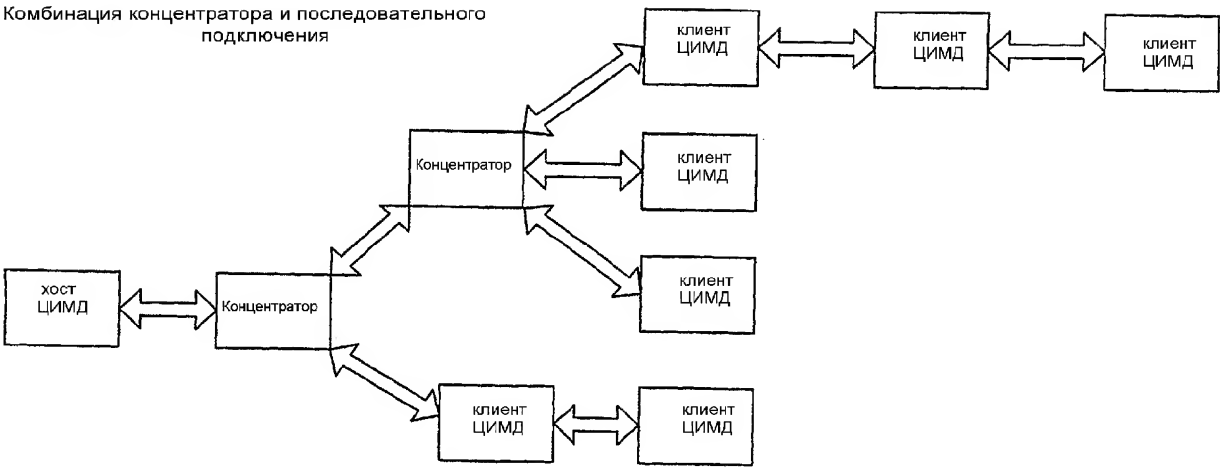
Каждый клиент повторяет данные, после того как ему был
присвоен адрес



Клиенты, подсоединенные через
концентратор

ФИГ.95

Комбинация концентратора и последовательного подключения



ФИГ.96

Форматирование данных кадры цветов

Пример: 6 битов Синего цвета, 8 битов Зеленого цвета, 7 битов Красного цвета



ФИГ.97